Fachhochschule Köln
Cologne University of Applied Sciences

Kölner Beiträge zur Technischen Informatik

Ausgabe-Nr. 2 / Januar 2012

ISSN 2193-570X

Herausgeber:

Rainer Bartz
Gregor Büchel (Redakteur)
Andreas Grebe
Georg Hartung
Hans W. Nissen
Lothar Thieling
Carsten Vogt

Unter Mitwirkung von:

Sebastian Seegert

Impressum:

Forschungsschwerpunkt Verteilte und Mobile Applikationen
Fachhochschule Köln
Fakultät für Informations-, Medien- und Elektrotechnik
Betzdorfer Str. 2
D-50679 Köln
gregor.buechel@fh-koeln.de
Stand: Januar 2012

Inhalt

Editorial……4

F. Shamim

Analyse von Testmodellierung und Testmethoden mit den Testwerkzeugen für ein existierendes komponenten-basiertes System……5

R. Erdmann, G. Hartung und T. Krawutschke

Hochzuverlässige eingebettete Systeme unter radioaktiver Strahlung……...11

O. Portugall

Objektive und subjektive Dienstegütecharakteristiken bei VoIP und IPTV……..17

S. Küffner

Vergleichende Analyse von Container- und Codectypen für IP- Videostreaming in IPTV-, WebTV- und Internet- Dienste……...28

B. Mainka

Extraktion und Analyse von VoIP-Daten zur Identifikation von Telefon-SPAM…….37

Editorial

Die vorliegende Ausgabe der „Kölner Beiträge zur Technischen Informatik“ enthält Beiträge zum Workshop des Forschungsschwerpunkts „Verteilte und Mobile Applikationen“, der am 13. Januar 2011 in Köln stattfand. Die Beiträge sind Forschungsarbeiten aus den Bereichen:

- Testmodellierung für komplexe Softwaresysteme,
- Entwicklung hochzuverlässiger eingebetteter Systeme,
- Dienstgütemessung und Codectypen bei VoIP, IPTV und WebTV,
- Identifikation von Telefon- SPAM in VoIP Netzwerken

Rainer Bartz

Gregor Büchel

Andreas Grebe

Georg Hartung

Hans W. Nissen

Lothar Thieling

Carsten Vogt

Analyse von Testmodellierung und Testmethoden mit den Testwerkzeugen für ein existierendes komponenten-basiertes System

Farhan Shamim

Abstract

Im Rahmen des Projekts wurden Erkenntnisse in der Anwendung relativ neuer Test-Standards gesammelt. Hierbei werden insbesondere UTP und TTCN-3 betrachtet. Diese Erkenntnisse sollen bei zukünftigen Arbeiten helfen, die richtigen Teststrategien auszuwählen. Dabei wird ein komplexes komponentenbasiertes System ausgewählt, das bereits in UML spezifiziert wurde. Für dieses System wird dann eine Testfallmodellierung mit UTP und Testfallprogrammierung mit TTCN-3 untersucht. Als ein komponentenbasiertes System wurde hierfür das Telefonnetzsimulation (TLS) System ausgewählt.

1 Problemstellung

Es werden mehrere System- und Integrationstestfälle erstellt, die die grundlegende Funktionalität des TLS-Systems überprüfen und das erwartete Verhalten der Software verifizieren. Diese Testfälle werden dann mithilfe von UML Testing Profile (UTP) modelliert. Zur Modellierung wird ein Werkzeug ausgewählt, welches insbesondere die Verfügbarkeit von Open-Source-Werkzeugen prüft. UTP ist zur Realisierung von modellgetriebenen Testens (MDT= Modell Driven Testing) entwickelt worden und der MDT-Aspekt wird deshalb bei der Auswahl eines Werkzeugs berücksichtigt. Da UTP auf UML basiert, sollen bestehende UML-Diagramme in möglichst großem Umfang wiederverwendet werden. In UTP modellierte Testsystem-Architektur kann dann TTCN-3 implementiert werden. Zur Implementierung des Testsystems ist TTWorkbench© von Testing Technologies GmbH einzusetzen. In TTWorkbench soll die Anbindung der Komponenten und das TLS-Systems durchgeführt werden. Bei der Erstellung von Testfällen wird ein verhaltensorientierter Verifikation-/Entwicklungsansatz (BDD) angewendet. Am Ende sollen dann die folgenden Fragen geklärt werden:

- Existieren (Open-Source) Werkzeuge, die eine benutzerfreundliche Modellierung in UTP ermöglichen?
- Wie hoch ist der Zusatzaufwand für den Systementwickler, um eine aussagekräftige Testmodellierung neben der Systemspezifikation zu erstellen?
- Ist die Umsetzung der Testfälle in TTCN-3 für die Struktur und Größe des Beispielsystems möglich und sinnvoll?
- Ist eine Automatisierung bei der Modellierung und bei der Implementierung möglich oder sogar bereits verfügbar?
- Welche zusätzlichen Aussagen über das SUT sind durch die verhaltensorientierte Verifikation mit TTCN-3 möglich oder ist ein zusätzliches Framework unbedingt erforderlich?

2 Komponenten in TLS-System

Das **Bild 1** stellt alle Komponenten des TLS-System mit ihren Abhängigkeiten dar. Alle Komponenten außer „Charging“ und „ISServer“ wurden bereits implementiert. Die Komponenten des TLS- Systems, außer „TelefonService“, „NetzGUI-Service“ und Netzmaster, realisieren hier die „IActivate-Component“-Schnittstelle.

Die Beziehungen für den „TelefonServiceProvider“ sind in **Bild 1** dargestellt. Die „IActivate-Component“-Schnittstelle dient zur Aktivierung und Deaktivierung von Komponenten für ein bestimmtes Netz. Somit kann die „Netzmaster“-Komponente alle Komponenten, die diese Schnittstelle implementieren, aktivieren und deaktivieren. Der „Netzmaster“ ist somit die „Main“-Komponente, die für das Starten und die richtige Installation der anderen Komponenten zuständig ist. Der „Netzmaster“ stellt auch eine GUI zur Verfügung (ein Netz-Editor bzw. Netz-Simulator), die von anderen Komponenten über die „INetzGUI“-Schnittstelle aktualisiert werden kann. Der „TelefonServiceProvider“ implementiert die „IGet-Telefon“-Schnittstelle, die durch den „Telefon-Service“ angeboten wird. Diese lose Kopplung ist notwendig, um die zyklische Abhängigkeit zwischen „TelefonService“ und „Vermittlungen“ aufzulösen. Zudem wird eine GUI für das Telefon vom „TelefonServiceProvider“ zur Verfügung gestellt.

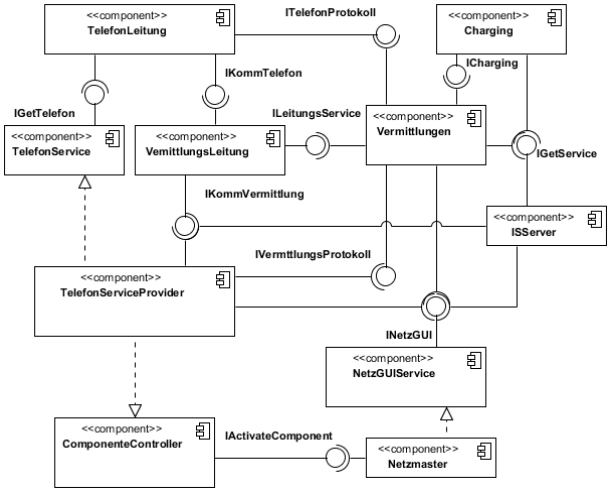

Bild 1 Komponente des Telefon Systems

3 Testmodellierung mit UML Testing Profile(UTP)

Für die Tool-Unterstützung wurden alle frei verfügbaren Tools, die auf den Seiten [2] und [3] gelistet sind, analysiert. Bei der Analyse wurden folgende Schritte durchgeführt:

1. Installation des Tools auf einem PC mit Windows 7 (x64) Betriebssystem
2. Modellierung eines in der UTP-Spezifikation aufgeführte Beispiels (Bank-Automat System)
3. Testfall-Generierung das in Schritt 2 modelliertem Beispiels

Fazit: Es wurde kein kostenloses Tool gefunden, welches offiziell das UTP unterstützt, d. h. bei den analysierten Tools ist das UTP mit anderen UML Profilen, wie EJB UML Profil, nicht aufgeführt und wird somit nicht unterstützt. Deshalb konnten nicht alle Konzepte aus UTP wie Testdaten-, Testverdicts-, Emulators-Konzept usw. modelliert werden. Allerdings ist es möglich, alle UTP-Stereotypen und Interfaces in diesen Werkzeugen zuerst manuell zu realisieren und dann gemäß UTP-Spezifikation zu modellieren. Die UTP spezifische Symbolen wie „Validation Actions“ und „Emulatoren“ können hier wiederum nicht modelliert werden. Wegen der fehlenden Unterstützung von UTP war hier die Testfall-Generierung nicht möglich. Wenn aber diese auch in einen MDD fähigen Tool UTP unterstützen würden, ist eine vollständige Testfall-Generierung nur in TTCN-3 möglich. Denn das komplette Mapping der UTP-Konzepte existiert nur für TTCN-3 (siehe [6]).

Nach weiterer Recherche wurden zwei professionelle kostenpflichtige Tools für UTP gefunden.

1. TTModelar für TTWorkbench von Testing Technologies
2. IBM Rational

Diese beiden Tools fokussieren sich auf die Kommunikationssysteme [5] und [6]. Im Rahmen dieses Projekts sollten jedoch nur die Testfälle für ein komponentenbasiertes System modelliert werden. Die genannten Tools kamen somit nicht zum Einsatz. Zudem sollten nur Open-Source- oder frei verfügbare Tools bei der Testfallmodellierung verwendet werden.

3.1 Testfallmodellierung mit UTP entfällt

Laut [5] ist das UTP nur so gut wie die Tools, welche das UTP unterstützen. Somit ist mit der jetzigen verfügbaren, kostenlosen Tool-Unterstützung des UTP kein modellbasiertes Testen mit UTP möglich.

Darüber hinaus, wenn das Testen allgemein in zahlreichen Open-Source Projekte im Betracht gezogen wird, gibt es ein Paradigmenwechsel in der Softwareentwicklung und im dem Testen von Software. Eine traditionelle Software–entwicklung (Analyse, Design, Code, Test) ist immer seltener zu sehen. Im Gegensatz hierzu wird die agile Softwareentwicklung wie Scrum, Kanaban, Lean, XP usw. immer populärer und häufiger eingesetzt. Testgetriebene Entwicklung (TDD und ATDD) und ihre zukünftigen Versionen der verhaltensorientierten Entwicklung (BDD) sind überall (meistens in USA, England, Australien und ein schnell wachsender Trend in Europa) bei großen und kleinen Projekten zu sehen. Die Softwareprojekte, die auf eine agile Softwareentwicklung setzen, zeigen, dass hochqualitative Produkte schneller und günstiger in dem Markt gebracht werden können. Es wurde aber kein Softwareprojekt (im Bereich Informationssystem) gefunden, das das UTP für das Testen einsetzt. Die Recherchen wurden bei großen Project-hosting Portalen wie Sourceforge, Google Code und Github durchgeführt. Ein möglicher Grund ist, dass das UTP bei der Testmodellierung das SUT benötigt. Hierbei werden die agilen Softwareentwicklungsansätze für Testfälle entwickelt, noch bevor eine SUT existiert. Ein Vergleich zwischen traditioneller und agiler Softwareentwicklung ist jedoch nicht Gegenstand dieses Artikels. Ziel dieser Arbeit ist es zu zeigen, dass der Erfolg des UTP abhängig von der Anpassung des UTP-Konzeptes an den Mainstream- Software-entwicklungsansatz ist.

Weiterhin ist es notwendig, dass das UTP-Tool „Reverse-Engineering“ und den „Round-Trip“ Mechanismus unterstützt, d. h. es muss nicht unbedingt zuerst modelliert und dann codiert werden. Denn in der agilen Softwareentwicklung werden zuerst Testfälle geschrieben

und dann codiert. Die Modellierung wird meistens durch „Reverse-Engineering“ gemacht, sobald die Quell-codebasis komplexer wird. Nun kann die Architektur des Quellcodes verbessert und durch „Round Trip“-Verfahren refaktorisiert werden, denn zu diesem Punkt existieren bereits die Testfälle und können ausgeführt werden. Der grüne Balken (des Test-Driven-Development Konzeptes) stellt sicher, dass das Refactoring keine Testfälle abgebrochen hat und die Software einen sog. „Test Driven Design“ oder einer „Test driven Architecture“ folgt.

4 System- und Integrationstestfälle mit TTWorkbench in TTCN-3

4.1 Telefon A ruft Telefon B an aber Telefon B antwortet nicht (SZ1)

- Gegeben: es existieren Telefone A 1234 und Telefon B 4321 und beide sind frei.
- Wenn Telefon A Telefon B anruft, dann sollte es bei Telefon B klingeln und bei Telefon B der Text "1234 ruft an..." und bei Telefon A der Text "Es wird bei 4321 geklingelt" angezeigt werden.
- Wenn Telefon B nicht antwortet, dann sollte nach 6 Sekunden die Verbindung abgebaut und bei Telefon A der Text "Kein Antwort von 4321" angezeigt werden.
- Nach 5 Sekunden sollte die Anzeige von Telefon A zurückgesetzt werden.
- Am Ende sollten beide frei sein.

4.2 Ausführung des Testfalls (SZ1)

Vor der Ausführung muss das TTCN-Testsystem mithilfe von dem SUT-Testadapter, also die „TLSSystem Adapter“- Klasse, an das SUT angebunden werden. Die Anbindung des TTCN-Testsystem an das SUT ist nicht trivial, wie es in Referenzbeispielen immer dargestellt wird. Vor allem deswegen, weil in bekannten Referenzbeispielen immer ein SUT gewählt wurde, welche separat läuft und per Sockets angesprochen werden kann. Im Rahmen dieses Projekts wurde ein komponentenbasiertes System getestet, indem keine Sockets verwendet wurden. Zur Kommunikation stellen die Komponenten des TLS-Systems (SUT) eine definierte öffentliche Schnittstelle zur Verfügung (siehe **Bild 1**), die von den Komponenten der „ServiceLoader“-Klasse angebunden werden. Über diese Schnittstellen erfolgt dann der eigentliche Informationsaustausch zwischen den Komponenten des TLS-Systems.

Darüber hinaus muss das richtige Hochfahren von SUT, im Gegensatz zu den Referenzbeispielen, in unserem Fall auch das TTCN-Testsystem übernehmen. In den Referenzbeispielen wurde das richtige Hochfahren von SUT nicht von dem TTCN-Testsystem übernommen, da die Kommunikation zwischen TTCN-Testsystem und SUT über Sockets stattgefunden hat. In diesem Fall muss das TTCN-Testsystem, in dem sich die Testfälle befinden, das SUT richtig hochfahren, alle von den Testfällen benötigten Komponenten aktivieren und die Testfälle bzw. BDD-Szenarios ausführen. Anschließend fährt das TTCN-Testsystem das SUT korrekt herunter, damit die Threads nicht zu „Waisen“ werden. Um dies erfolgreich zu lösen, muss das TTCN-Testsystem das SUT mit den Bibliotheken in den eigenen Klassenpfad aufnehmen. Das TTCN-Testsystem wurde mit der TestingTech Workbench entwickelt. Die Entwicklungsumgebung ist in der Lage, den Klassenpfad des TTCN-Testsystems für das SUT so anzupassen, dass das SUT richtig hoch- und herunterfahren kann. Auch nach weiteren Untersuchungen wurden keine weiteren Möglichkeiten gefunden, in der der Klassenpfad von TestingTech Workbench verändert werden kann. Es gibt zwar eine Variante, in der zusätzliche Argumente an die JVM eingegeben werden können, jedoch kann der Klassenpfad nicht geändert werden. Die richtige Festlegung des Klassenpfads ist wichtig, denn das TTCN-Testystem und das SUT müssen im gleichen Prozess laufen, da die Kommunikation zwischen den Komponenten des SUTs nicht prozessübergreifend realisierbar ist. Ist dies nicht gegeben, kann die „ServiceLoader“-Klasse die gewünschten Schnittstellen nicht laden, denn sie sucht die Implementierungen der Schnittstellen im Klassenpfad. Wurde keine Implementierung von gewünschten Schnittstellen gefunden, wird die „NoSuchElementException“ ausgelöst. Diese Einschränkung wurde aus der Spezifikation [4] der „ServiceLoader“-Klasse entnommen. Die einzige Möglichkeit die besteht, ist den Klassenpfad in die „manifest.mf“ Datei der TLSTestadapter-Komponente einzutragen, in der die „TLSSystemAdapter“-Klasse hinterlegt ist. Dieser Klassenpfad kann nur dann aus der „manifest.mf“-Datei gelesen werden, wenn das System mit dem Befehl „java –jar EinSystem.jar“ gestartet wurde. So konnte das TTCN-Testsystem teilweise an das SUT gebunden werden. Zudem liegt es daran, dass wenn die SUT den Versuch startet, eine Schnittstelle mithilfe der „ServiceLoader“-Klasse anbindet, eine „NoSuchElementException“-Exception ausgelöst wird. Der Grund ist hier, dass der Klassenpfad nicht die veröffentlichten Implementierungen der Schnittstellen enthält.

Als Alternative wurden alle abhängigen JAR-Dateien des TLS-Systems in das JDK-Verzeichnis hinein kopiert, somit in den globalen Klassenpfad, der für alle Java-Anwendungen gültig ist. So ist es möglich, alle veröffentlichten Schnittstellen mithilfe der „ServiceLoader“-Klasse zu finden. Mit dieser Strategie kann das TTCN-Testsystem an das TLS-System angebunden werden, jedoch muss beim Laden einer Schnittstelle der richtige „ClassLoader“ angegeben werden. Um diesen Fall von „ClassLoader“ zu untersuchen, wurde ein Testfall geschrieben, in dem eine „IHelloService“-Schnittstelle mithilfe von der „Service-Loader“-Klasse angebunden wird. Die „IHello-Service“-Schnittstelle wurde in der „HelloService“-Komponente definiert und in der „HelloServiceProvider“- Komponente implementiert sowie veröffentlicht. Im normalen Fall (eine veröffentliche Implementierung einer Schnittstelle), wenn der Klassenpfad richtig gesetzt ist, ist die Anbindung der „IHelloService“-Schnittstelle wie folgt definiert:

```
IHelloService s =
ServiceLoader.load(IHelloService.class).iterator.next
();
```

Jetzt liegt der Referenz der veröffentlichten Implementierung von „IHelloService“, die von „HelloServiceProvider“-Komponente zur Verfügung gestellt wurde, in der Variablen „s“. Die oben angegebene Anweisung meldet „NoSuch ElementException“, da die veröffentliche Implementierung der „IHelloService“-Schnittstelle in dem Klassenpfad durch den „ServiceLoader“ nicht gefunden wurde. Aber es existieren nun alle Komponenten im globalen Klassenpfad. Um dieses Problem zu lösen muss dem ServiceLoader der „ClassLoader“ der Schnittstelle explizit angegeben werden. Dies wird wie folgt definiert:

```
IHelloService s =
ServiceLoader.load(IHelloService.class,IHelloService.
class.getClassLoader());
```

Nun kann mit der obigen Anweisung die richtige Implementierung der „IHelloService“- Schnittstelle geladen werden. Im SUT wurden die Schnittstellen immer ohne den ClassLoader geladen, deswegen muss das SUT jetzt angepasst werden damit die erste Ausführung überhaupt möglich ist. Demzufolge wurde das SUT angepasst und es werden alle Schnittstellen immer mit dem zugehörigen ClassLoader geladen. Jetzt kann das System richtig hochgefahren werden.

Bei der Ausführung des ersten Szenario (SZ1) wurde das SUT zwar richtig hochgefahren aber es trat ein anderes Problem auf. Das SUT konnte nicht auf die Datenbank zugreifen. Die Exception, die bei der Ausführung ausgelöst wurde, heißt „PersistenceException“, was bedeutet, dass keine „PersistenceUnit“ gefunden wurde. Im SUT wird auf die Datenbank mithilfe der Java Persistence API (JPA 2.0) und der zugehörigen Implementierung „EclipseLink“ zugegriffen. Um das Problem zu verstehen wird zuerst der Aufbau von JPA mit „EclipseLink“ kurz erläutert. Die JPA ist eine von der Firma Sun (jetzt Oracle) spezifizierte Menge an Schnittstellen, die den Datenbankzugriff vereinheitlichen. JPA bietet den Service, unterschiedliche Datenbanken einheitlich anzusprechen ohne dabei JDBC spezifisch programmieren zu müssen. Es gibt viele Service-Provider, die JPA-Services implementieren. In dieser Arbeit wurde zum Beispiel eine von vielen Opensource-Implementierungen „EclipseLink“ eingesetzt. Im Grunde genommen werden die Implementierungen der JPA-Schnittstellen auch durch den ServiceLoader-Mechanismus geladen und verwendet. Also, muss hier auch der „ServiceLoader“ und der „ClassLoader“ explizit angegeben werden, damit die „Persistence-Unit“ gefunden werden kann. Die Durchführung dieser Änderung in der JPA „EclipseLink“-Implementierung ist nicht sinnvoll.

Dieses Ergebnis stellt die Frage, ob das TTCN-Testsystem weiterhin für das Testen des TLS-Systems eingesetzt werden kann und ob dieses sinnvoll ist. Die Schlussfolgerung, dass das Testen eines komponenten-basierten Systems mit TTCN-3 nicht funktioniert, ist selbst ein Ergebnis. Für zukünftige Arbeiten sind diese Erkenntnisse der technischen Randbedingungen und Eigenschaften des SUTs jedoch zu dokumentieren, da es sehr aufwendig oder nahezu unmöglich ist, das Testen mit dem TTCN- Testsystem durchzuführen.

4.3 Ausscheidungskriterien für TestingTech Workbench

4.3.1 Verwendung von „ServiceLoader“-Mechanismus

Wenn ein SUT aus mehreren Komponenten besteht und die Kommunikation untereinander über den „Service-Loader“-Mechanismus läuft, dann bietet die jetzige Version von TestingTech Workbench noch nicht die Möglichkeit ein solches SUT an ein TTCN-Testsystem anzubinden. Das gilt natürlich auch für die Bibliotheken die das SUT verwenden sollte.

4.3.2 Datenbankzugriff mit Java Persistence API

In dieses Projekt wurde z. B. JPA verwendet, die auf den „ServiceLoader“-Mechanismus basiert.

4.3.3 SUT mit GUIs

Wenn ein SUT mehrere GUIs hat, fährt TestingTech Workbench das SUT nach dem Hochfahren nicht richtig herunter. Das Herunterfahren des SUTs muss dann manuell gemacht werden.

4.3.4 Kein auf Sockets-basierter Mechanismus der als Kommunikationsschnittstelle mit dem SUT dient

Wenn ein SUT zur Kommunikation keine Sockets anbietet, dann ist der „Message-based Communication"-Ansatz, um das SUT zu testen, sehr aufwendig, fehleranfällig und nicht entwicklerfreundlich.

Während dieses Projekts hat sich ein Punkt allgemein immer wieder hervorgehoben, und zwar, dass TTCN-3 bzw. TestingTech Workbench eventuell gar nicht dafür ausgelegt ist, Systeme zu testen, die wir testen wollen. Wenn man eine Mücke mit einer Pistole erschießen möchte, muss man sehr gut mit der Pistole schießen können oder man greift doch zur einfachen Fliegenklatsche. Sie ist billiger und hält sich besser. Das richtige Werkzeug ist wichtig um ein Problem möglichst effektiv zu lösen.

4.3.5 Ein bestimmtes Protokoll sprechendes SUT

TestingTech Workbench lässt sich besonders gut einsetzen, wenn getestet werden soll, ob das SUT ein bestimmtes Protokoll beherrscht. Hier zeigt TTCN-3 seine Stärken. Mithilfe der TTCN-3 Sprachkonstrukte wie „alt, send, receive, timers usw." können die Testfälle für ein Protokoll sehr leicht und verständlich erstellt werden, da sich Sequenzdiagrammen sehr einfache in TTCN-3 umsetzen lassen. Z. B. wenn ein SUT behauptet, dass es das SIP-Protokoll beherrscht, dann braucht nur noch der Testadapter für das SUT beschrieben werden. Denn die Testfälle zum Testen des SIP-Protokolls brauchen nur einmal geschrieben zu werden. So eine Testsuite kann für viele SUTs eingesetzt werden. Dies erspart enorm viel Zeit in der Testphase.

5 Alternative Tools

Es wurden folgenden populären open source Testing tools ausgewählt und mithilfe dieser Tools alle Testszenarien implementiert und das Ergbenis mit UTP und TTWorkbench© Resultate verglichen.

5.3 FitNesse

FitNesse ist eine Webanwendung in der ein Webserver, ein Wiki-System und ein automatisiertes Testwerkzeug integriert sind. FitNesse basiert auf Ward Cunninghams's Framework for Integration Test. FitNesse ist zur Automatisierung und Unterstützung von Abnahmetests und Integrationsstests von Robert C. Martin entwickelt worden. Mit FitNesse können nicht nur Tester und Entwickler gemeinsam auf einer Platform kollaborieren, sondern auch nicht technische Projektmitglieder wie Kunden können die Projektfortschritte nachvollziehen. FitNesse braucht keine Konfiguration oder Installation, da es als Webanwendung implementiert ist. Man startet einfach FitNesse, indem man die „fitnesse.jar" Datei ausführt und den Browser auf „localhost:8080" zeigen lässt [8].

5.4 Cucumber

Cucumber ist auch ein Testautomatisierung-Framework für Integration- und Abnahmetests. Cucumber beschreibt allerdings die Szenarien streng nach BDD-Prinzipien. Die Szenarien werden in einer Text-Datei mit der Endung „feature" gespeichert. Der Fixture-Code muss jedoch mit der dynamischen Skriptsprache „Ruby" erfolgen. Im Gegensatz zu FitNesse liegen die Szenarien und zugehörige Fixture-Code in der gleichen Entwicklungsumgebung, was den Vorteil mitbringt, dass die Szenarien und Fixture-Code refaktorisiert werden können. Das ist bei einer großen Anzahl der Szenarien sehr hilfreich und ein „Killer"-Feature. Der Nachteil vom Cucumber ist, dass der Fixture-Code in einer anderen Sprache als die Implementierungssprache verfasst ist. Weitere In-formationen können über die Homepage [7] von Cucumber abgerufen werden.

6 Schlussfolgerung und Ausblick

UTP ist ein mächtiges Konzept und könnte die Antwort auf die Probleme der komplexen Testsysteme sein. Zurzeit gibt es allerdings zu wenig Toolunterstützung, die für das Model-Driven Testing nötig ist. Darüber hinaus muss UTP um folgende Punkte erweitert werden:

- Fokus auf Informationssysteme und Web-anwendungen
- Vollständige UTP Mappings für anderen Frameworks außer TTCN-3
- Hierarchische Datenkonzepte
- Anpassung zum agilen Entwicklungsprozess

TTCN-3 konnte sich im Bereich des Testens von Kommunikationssystemen behaupten und wird immer mehr beim Testen von eingebetteten Systemen eingesetzt. In diesem Projekt wurde untersucht, ob TTCN-3 auch für das Testen von Informationssystemen geeignet ist. Was mit Sicherheit gesagt werden kann ist, dass der Message-based Communication Ansatz von TTCN-3 nicht der

richtige Ansatz für das Testen solcher Systeme ist. Die TestingTech Workbench muss mehr Unterstützung zum Testen normaler Informationssysteme anbieten. Zusätzlich lässt TTCN-3 sich besonders gut in Projekten einsetzen, in denen das traditionelle Wasserfallmodel angewendet wird, d. h. wo erst am Ende getestet wird oder vor den Testfällen bereits ein SUT existiert, wohingegen beim agilen Entwicklungsprozess schon vor der Implementierung eines SUTs die Testfälle erstellt und ausgeführt werden können. TTCN-3 muss also auch an den agilen Entwicklungsprozess angepasst werden.

Andere Frameworks wie FitNesse und Cucumber werden immer populär und häufiger eingesetzt. Diese sind speziell zum Testen von Informationssystemen entwickelt worden. Die Entwickler-Community dieser Frameworks ist ebenfalls sehr aktiv und bietet Einsteigern eine gute Hilfestellung.

7 Literatur

[1] http://www.nt.fh-koeln.de/fachgebiete/inf/nissen/sif/Liu_UTP_Bericht.pdf (aufgerufen 06.09.2010)

[2] http://www.objectsbydesign.com/tools/umltools_byCompany.html (aufgerufen 06.09.2010)

[3] http://www.omg.org/technology/documents/formal/test_profile.htm (aufgerufen 06.09.2010)

[4] http://download.oracle.com/javase/6/docs/api/java/util/ServiceLoader.html (aufgerufen 29.09.2010)

[5] Colin Willcock, Thomas Deiß, Stephan Tobies, Stefan Keil, Federico Engler, Stephan Schulz:An Introduction to TTCN-3,

[6] Paul Baker, Zhen Ru Dai, Jens Grabowski, Oystein Haugen, Ina Schieferdecker, Clay Williams: Model-Driven Testing Using the UML Testing Profile, Springer Verlag

[7] http://cukes.info/ (aufgerufen 29.09.2010)

[8] http://fitnesse.org/FitNesse.UserGuide.OneMinuteDescription (aufgerufen 29.09.2010)

8 Autor

M. Sc. Farhan Shamim, studiert im Masterstudiengang „Technische Informatik“ und ist wissenschaftlicher Projektmitarbeiter am Institut für Nachrichtentechnik der Fachhochschule Köln.

Kontakt: farhan.shamim@fh-koeln.de

Hochzuverlässige eingebettete Systeme unter radioaktiver Strahlung

Ralph Erdmann, Georg Hartung und Tobias Krawutschke

Abstract

Im Detector Control System (DCS) Projekt der Fakultät für Nachrichtentechnik an der FH Köln wird das hochzuverlässige Detector Control System Board (DCSB) entwickelt. Das DCSB kommt beispielsweise am ALICE Beschleunigerexperiment am CERN zum Einsatz (siehe [1]). Da die bisher verwendete CPU mittlerweile nur noch schwer bis gar nicht erhältlich ist und zudem von einem DCSB in neuen Teilchenbeschleunigern weitergehende Funktionalität erwartet wird, muss für zukünftige Teilchendetektoren eine neue Generation von DCSBs entwickelt werden. Um die Entscheidung für den zentralen Prozessor- und FPGA-Chip (Field Programmable Gate Array) des neuen DSCB zu unterstützen, wurde in diesem Experiment ein FPGA vom Typ Actel A3P-1000 bestrahlt. Das Ziel des Experimentes ist die Bestimmung der Eignung des Bauteils zum Einsatz in strahlenkritischen Bereichen durch Berechnung des Wirkungsquerschnitts.

1 Fehlerarten und Korrekturmaßnahmen bei FPGAs in radioaktivem Umfeld

Fast jede Art von Strahlung und Teilchen kann beim Durchqueren von Materie Auswirkungen nach sich ziehen. Das ionisierende Teilchen gibt beim Durchqueren Energie an das umliegende Material ab, die als Linear Energy Transfer (LET) bezeichnet wird. Diese Änderung der Ladungsverteilung kann ungewollte Effekte auslösen, die man in Hard und Soft Errors unterteilen kann. Die Gruppe der Hard Errors beschreibt an Hardware entstehende Schäden, die nicht korrigiert werden können, wie z.B. die Zerstörung eines MOS-Transistors durch *latch up*, ausgelöst durch ein mit hoher elektrischer Energie geladenes Teilchen. Sie treten bei Anwendungen in der Raumfahrt oder bei Experimenten an Teilchenbeschleunigern selten auf, da die Energie der auftretenden Strahlung zu gering ist. Soft Errors treten bereits bei schwächerer Strahlung auf. Man unterscheidet zwischen dem *single event upset (SEU)* und dem *single event transient (SET)*.

Der SEU ist der in diesem Umfeld „wichtigste“ Soft Error: Er stellt das „Kippen“ einer Speicherschaltung in den inversen Zustand durch die Ladung eines Teilchens dar (engl.: *bitflip*) und kann durch Neukonfiguration der Schaltung behoben werden. Der SET ähnelt dem SEU und bezeichnet den temporären Zustandswechsel einer kombinatorisch arbeitenden Schaltung (engl.: *glitch*). Der SET wird als Fehlfunktion nur wirksam, wenn er in der Nähe einer aktiven Taktflanke auftritt und zur Speicherung eines falschen Werts führt.

Die genannten Soft Errors sind durch geeignete Fehlertoleranzverfahren korrigierbar. Ein häufig eingesetztes Verfahren ist Triple Modular Redundancy (TMR). Hierbei wird ein einzelnes Modul verdreifacht und die drei Ergebnissignale werden an einen Voter geschickt. Dieser trifft dann eine Mehrheitsentscheidung, womit Fehler eines der drei Module korrigiert werden. (Für weitere Informationen siehe auch [2].)

2 Messung der Fehlerwahrscheinlichkeit für FPGAs

2.1 Wirkungsquerschnitt (Cross Section)

Der Wirkungsquerschnitt (engl.: *Cross Section*) ist in der Teilchenphysik ein Maß für die Wahrscheinlichkeit, dass zwischen einem Ziel und einem einfallenden Teilchen eine Interaktion (beispielsweise ein Soft Error) stattfindet. Er wird durch den griechischen Buchstaben σ (sigma) dargestellt (Einheit cm^2) (siehe auch [1]).

In den Wirkungsquerschnitt gehen ein:

- Anzahl der Fehler (n)
- mittlere Betriebsdauer zwischen Ausfällen (engl.: *meantime between failures – MTBF*)

$$MTBF = \frac{Zeit}{n}[s]$$

- Flussdichte (engl.: *flux*), die Anzahl der Teilchen (p), die eine bestimmte Fläche pro Sekunde durchfließen

$$flux = \frac{Teilchen}{Fläche \cdot Zeit}\left[\frac{p}{s \cdot cm^2}\right]$$

Daraus ergibt sich der Wirkungsquerschnitt zu:

$$\sigma = \frac{n}{MTBF \cdot flux} [cm^2]$$

Die Vorgehensweise zur Bestimmung von σ liegt auf der Hand: Man misst bei radioaktiver Strahlung die Anzahl der Fehler in einer bestimmten Zeit und erhält so n und MTBF. Bei bekannter Flussdichte kann man dann σ berechnen.

2.2 Der Flash FPGA Actel ProASIC3

Der Actel ProASIC3 A3P-1000 FPGA ist ein Flash-FPGA mit folgenden Eckdaten:

- 1.000.000 System Gates
- 24.576 VersaTiles
- 144 Kbits RAM
- 1024 bits FlashROM
- 1 PLL

Der FPGA basiert auf Flash-Speicher. Dieser ist im Gegensatz zu SRAM Zellen nicht flüchtig, verliert also seine Daten nicht wenn er ausgeschaltet wird. Flash-Speicher basiert auf Floating Gate Transistoren. Über dem Floating Gate befindet sich ein weiteres Control Gate. Durch Anlegen von Spannung an das Control Gate kann das Floating Gate geladen und entladen werden. Ist das Floating Gate einmal geladen, so bleibt dieser Zustand erhalten, egal ob der Speicher mit einer Versorgungsspannung verbunden ist oder nicht. Der Hauptvorteil von Flash FPGAs bezogen auf den Einsatz in strahlungskritischen Bereichen ist, dass das Floating Gate eine wesentlich höhere Spannung braucht um seinen Zustand zu wechseln, als ein einzelnes ionisierendes Teilchen abgeben kann. FPGAs dieser Technologie werden als weitestgehend resistent gegenüber SEUs im Konfigurationsteil betrachtet. Die Flip-Flops hingegen sind genauso anfällig für SEUs wie die Flip-Flops von FPGAs, die in SRAM Technologie gefertigt wurden. (Nach [3], siehe auch dort für weiterführende Informationen.)

2.3 Die Testschaltung: Schieberegister

Als Schieberegister bezeichnet man Schaltungen, die mehrstellige binäre Signale mit dem Takt aufnehmen, speichern und wieder abgeben können.

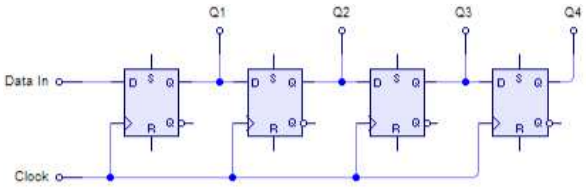

Bild 1 4 Bit Schieberegister mit D-Flip-Flops [4]

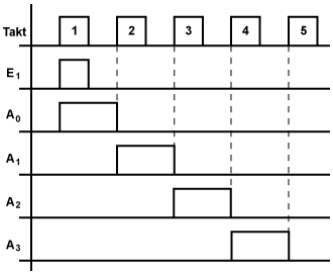

Bild 2 Impulsdiagramm Schieberegister [5]

Wird am Eingang eines Schieberegisters ein Signal (Data In) angelegt, so wird dieses mit jedem Takt (Clock) um eine Stelle (ein Flip-Flop) weiter gerückt. Um SEUs festzustellen wird der zu testende FPGA mit einem Schieberegister programmiert, welches den gesamten Speicher des Bauteils einnimmt. Über DATA_IN wird ein Bitmuster in den FPGA geladen und anschließend wieder ausgelesen. Es wird verglichen ob die ausgelesenen Daten den erwarteten entsprechen. Kommt es zu einer Abweichung, so kam es zu einem bitflip während dem Durchlauf durch den FPGA, d.h. ein SEU wurde festgestellt.

Um das Einspeisen und Auslesen des Bitmusters mit hoher Datenrate durchführen zu können, wurde eine Hardware mit FIFO Speicher im CPLD (Complex Programmable Logic Device) des DCS-Boards implementiert. Diese Hardware übernimmt auch die Taktung des Schieberegisters und kann über in C geschriebene Software gesteuert werden.

2.4 Das Development Board "A3P Devboard"

Der A3P-1000 befindet sich auf einem Development Board, im folgenden A3P Devboard oder nur Devboard genannt. Es wurde im Rahmen einer Diplomarbeit [6] an der Fachhochschule Köln entwickelt. Für das Experiment interessant sind die über Expansion Header von außen gut erreichbaren PINs des FPGA und die JTAG Schnittstelle (Joint Test Action Group), eine verbreitete Test und Debug Schnittstelle).

2.5 Versuchsaufbau

Um den FPGA zu testen wurde folgende Auslesekette entwickelt:

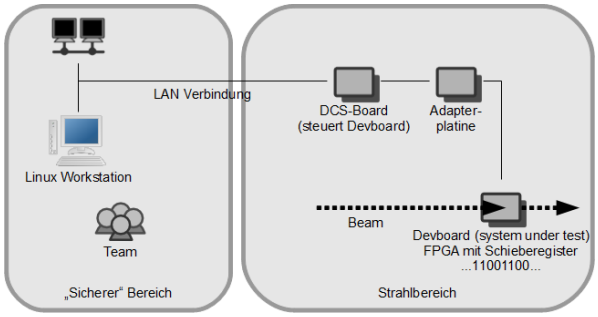

Bild 3 Auslesekette

Der FPGA befindet sich auf dem Devboard im Strahl ("Beam"). Ein DCS-Board steuert das Devboard mit den folgenden Funktionen:

- Lesen und Schreiben des FPGA auf dem Devboard
- Steuerung der Spannungsversorgung des Devboard
- Lesen und Schreiben der Konfiguration des FPGA über JTAG

Das DCS-Board ist über eine Netzwerkschnittstelle mit einer Linux-Workstation verbunden, über die das Experiment aus der Ferne überwacht und gesteuert werden kann; es dient dem DCSB zusätzlich als Speichermöglichkeit.

2.6 Testaufbau am COSY

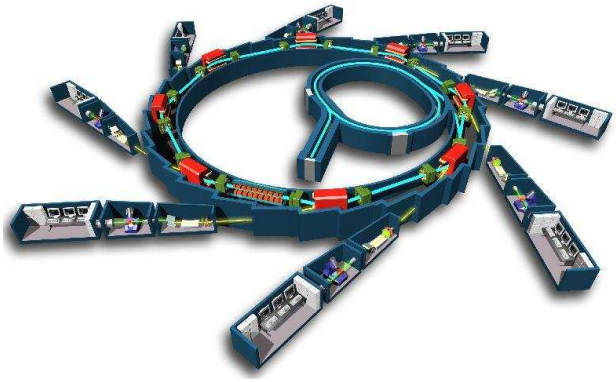

Bild 4 Beispielabbildung eines Synchrotron [7]

Der Cooler Synchrotron (COSY) ist ein Teilchen-beschleuniger mit Speicherring im Forschungszentrum Jülich. Der Beschleuniger hat einen Umfang von 184m (zum Vergleich: Der Umfang des LHC am CERN beträgt ca. 26km). Er wird, wie der LHC auch, für Experimente der Hadronenphysik eingesetzt.

Bild 5 Blick in die Cave

Bild5 zeigt die Cave, den Bereich, in den der Beam geleitet wird. Nach Durchqueren der Experimente trifft der Strahl auf den aus einer Metalllegierung bestehenden Beamdump, welcher die Teilchen des Beams absorbiert (nicht im Bild).

Bild 6 Testaufbau im Detail 1

Bild6 und Bild7 zeigen den Testaufbau in der Cave. Die Hardware wurde mit Aluminium-Profilen im Strahl positioniert.

- Platine Oben: Devboard mit Actel A3P-1000
- Platine Mitte: Adapterplatine
- Platine Unten: DCS-Board

(Vergleiche auch **Bild 3**: Auslesekette)

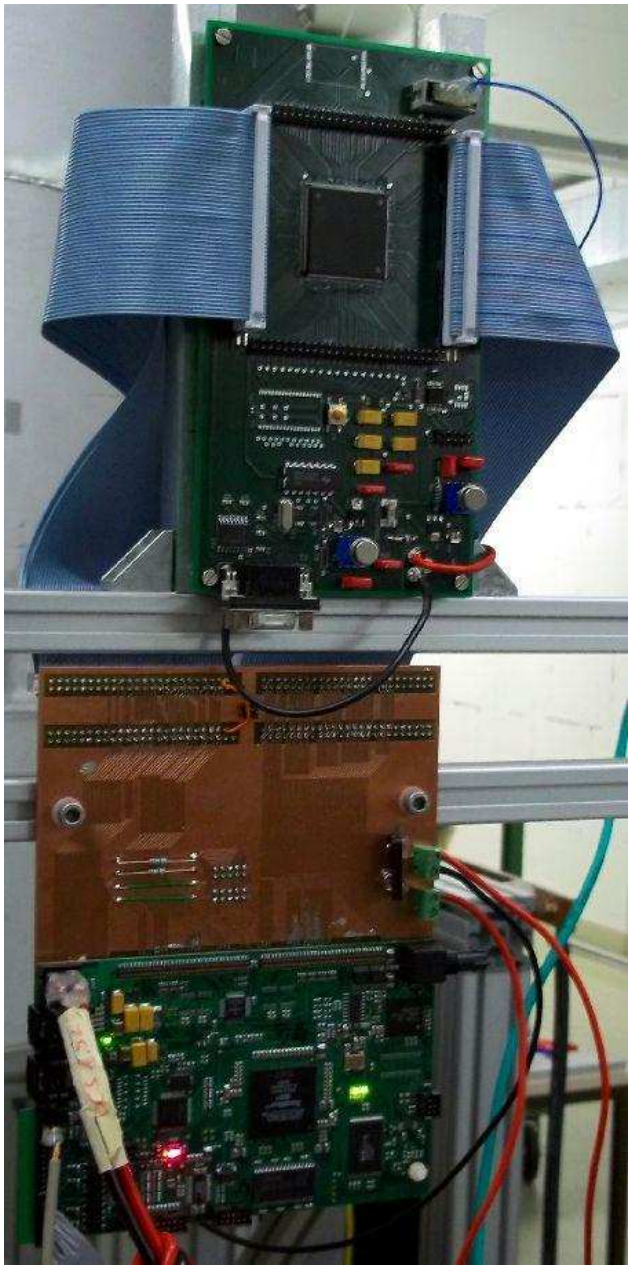

Bild 7 Testaufbau im Detail 2

Der eingespeiste Beam hat folgende Eigenschaften:

- 2cm Durchmesser
- besteht aus Protonen (ionisierter Wasserstoff)
- ca. 2GeV Intensität[1]
- ca. $3*10^9$ Teilchen pro 5 Minuten. Diese als Spill bezeichnete Angabe muss durch entsprechende Messgeräte im Betrieb überprüft werden, da aus diesen Werten die Flussdichte berechnet wird.

[1] Das Elektronvolt wird in der Atom-, Kern- und Teilchenphysik benutzt und bezeichnet die Energie von Teilchen. 1eV entspricht $1{,}602176487(40)*10^{-19}$ J(oule)

3 Auswertung der Messergebnisse

3.1 Probleme bei der Bestimmung der Flussdichte

Während der Durchführung des Experimentes traten Probleme auf. Die Teilchen wurden nicht gleichmäßig sondern in stark gepulster Form eingespeist.

Dies führte dazu, dass die Szintillatoren, welche zur Messung der Flussdichte eingesetzt wurden, nicht funktionierten, da sie die hohe Teilchenanzahl in den Pulsen nicht verarbeiten konnten. Ohne die Kenntnis der Flussdichte ist die Berechnung der Cross Section nicht möglich. Um dies zu kompensieren, musste die Flussdichte anhand der bekannten Cross Section anderer SUTs (System Under Test) im Experiment und deren gemessenen SEUs abgeschätzt werden. Daher sind die Ergebnisse der Messung eher qualitativ einzustufen.

3.2 Messdaten

In 7 Nächten mit Bestrahlung wurden insgesamt 147 SEUs festgestellt.

Run 34 als Beispiel für einen Testlauf :

- Mi., 15. Dezember, 01:33 bis 03:24
- es wurden 6 SEUs in VersaTiles festgestellt
- 5 flips von 1 auf 0
- 1 flip von 0 auf 1

In der gerundeten Laufzeit von 2 Stunden wurden 6 SEUs gemessen. Dies entspricht einer MTBF von 1200 Sekunden.

3.3 Cross Section Beispielrechnung

Wie bereits dargestellt ist es aufgrund der fehlenden Werte der Flussdichte noch nicht möglich genaue Rechnungen zur Cross Section durchzuführen. Ohne diese Werte ist es ebenfalls auch nicht möglich, die MTBF zu bestimmen, da nicht klar ist, wann genau das SUT dem beam ausgesetzt war.

Für eine Beispielrechnung kann man grob geschätzt die folgenden Werte annehmen:

- ca. $3*10^9$ Teilchen pro 5 Minuten
- Circa alle 1200 Sekunden ein Fehler (MTBF)
- Fläche des FPGA beträgt ca. 0,36 cm² (Für diesen Wert wurde ein baugleicher FPGA abgeschliffen und der Dye freigelegt. Dieser hat die Kantenlängen 0,06cm * 0,06cm. Allerdings wird

diese Fläche auch von anderen Komponenten als den ausgewerteten VersaTiles belegt.)

Mit den bereits vorgestellten Formeln lässt sich nun die Flussdichte des gesamten Strahls abschätzen:

$$flux_{strahlgesamt} = \frac{3 \cdot 10^9 \, p}{300s \cdot 1 \cdot \pi \cdot cm^2} \approx 3{,}2 \cdot 10^6 \, \frac{p}{s \cdot cm^2}$$

Anteile, die einen 0,36cm^2 IC treffen:

$$\frac{0{,}36 cm^2}{1 \cdot \pi \cdot cm^2} \approx 0{,}11$$

Daraus ergibt sich dann der flux im Bauteil zu:

$$flux_{chip} = 3{,}2 \cdot 10^6 \, \frac{p}{s \cdot cm^2} \cdot 0{,}11$$
$$\approx 0{,}36 \cdot 10^6 \, \frac{p}{s \cdot cm^2}$$

Nun lässt sich die Cross Section bestimmen:

$$\sigma = \frac{1}{1200s \cdot 0{,}36 \cdot 10^6 \cdot 1 \diagup s \cdot cm^2} \approx 2{,}3 \cdot 10^{-9} cm^2$$

Zum Vergleich: Beim "alten" DCS-Board wurden Cross Section Werte im Bereich von $\sigma = 1...7 * 10^{-9} \, cm^2$ berechnet. Der Wert der Beispielrechnung befindet sich in der gleichen Größenordnung, was die aufgenommenen Werte plausibel erscheinen lässt.

4 Ausblick

Die bereits genannten Probleme bei der Messung der Teilchendichte gestatten bisher nur eine näherungsweise – man könnte auch sagen qualitative – Berechnung des Wirkungsquerschnitts. Daher sollen zunächst die in den weiteren gleichzeitig durchgeführten Experimenten ermittelten Teilchendichte-Werte genutzt werden, um den Wirkungsquerschnitt exakter zu bestimmen.

Ein weiteres Ziel ist die Untersuchung der bisher noch nicht ins Experiment einbezogenen Komponenten PLL und RAM des FPGA. Dazu müssen komplexere Konfigurationen entwickelt werden, die analog zur Schieberegisterschaltung für die "VersaTile"-Zellen eine einfache Aussage über Fehler ermöglichen.

Neben diesen eher grundsätzlichen Betrachtungen ist aber auch eine Untersuchung an Zielarchitekturen erforderlich. Die Hardware-Architektur der FPGAs auf einem DCS-Board wird je nach Einsatzort aus simplen Zu–standsmaschinen bis hin zu komplexen Prozessor-Speicherkombinationen bestehen. Es ist daher er–forderlich, möglichst frühzeitig auch solche Architekturen bezüglich ihres Wirkungsquerschnitts zu berechnen, damit passende Maßnahmen für die Erhöhung der Fehlersicherheit getroffen werden können. Zuletzt sei in diesem Zusammenhang darauf hingewiesen, dass auch durch Softwaremaßnahmen unter Inkaufnahme einer höheren Bearbeitungszeit die Fehlertoleranz erhöht werden kann und daher auch der Effekt dieser Möglichkeiten durch Experimente bestimmt werden muss.

5 Literatur

[1] Krawutschke, Tobias: Reliability and Redundancy of an Embedded System used in the Detector Control System of the ALICE Experiment, Dissertation, Wiku Verlag, 2010

[2] Gebelein, Engel, Kebschull: FPGA Fault Tolerance in Particle Physics Experiments. In: it - Information Technology 4/2010, Oldenbourg Verlag, München

[3] Engel, Heiko: Development of a Fault Tolerant Softcore CPU for SRAM based FPGAs, Diplomarbeit, Universität Heidelberg, 2009

[4] Bildmaterial von der Webseite http://de.wikipedia.org, Artikel Schieberegister, letzter Abruf 13.03.2011

[5] Bildmaterial von der Webseite http://www.elektronik-kompendium.de, Artikel Schieberegister, letzter Abruf 13.03.2011

[6] Scholl, Benjamin: Entwicklung eines Development-Boards für einen Actel FPGA zum Test fehlertoleranter Systemarchitekturen, Diplomarbeit, Fachhochschule Köln, 2010

[7] Bildmaterial von der Webseite http://de.wikipedia.org, Artikel Synchrotron, letzter Abruf 13.03.2011

Autor

B. Sc. Ralph Erdmann, studiert im Masterstudiengang „Technische Informatik“ und ist wissenschaftlicher Projektmitarbeiter am Institut für Nachrichtentechnik der Fachhochschule Köln.

Kontakt: ralph.erdmann@fh-koeln.de

Dr. rer. nat. Tobias Krawutschke, ist wissenschaftlicher Mitarbeiter am Institut für Nachrichtentechnik der

Fachhochschule Köln und Mitarbeiter bei der European Organization for Nuclear Research CERN.

Kontakt: tobias.krawutschke@koeln.de

Prof. Dr.-Ing. Georg Hartung, ist Dozent für das Lehrgebiet „Technische Informatik“ mit dem Schwerpunkt „Eingebettete Systeme“ am Institut für Nachrichtentechnik der Fachhochschule Köln.

Kontakt: georg.hartung@fh-koeln.de

Objektive und subjektive Dienstegütecharakteristiken bei VoIP und IPTV

Oliver Portugall

Abstract

Während einer Übertragung von Video- oder Audiostreams über ein Netzwerk sind Störungen und somit Einbrüche der Qualität möglich. Um diese messen und beurteilen zu können sind äußerst komplexe Verfahren nötig, da neben objektiven Messparametern besonders subjektive Empfindungen Auswirkungen auf die Qualitätseinschätzung haben.

In dieser Arbeit werden unterschiedliche Ansätze zur Messung einer subjektiven Qualität vorgestellt und deren Unterschiede dargestellt. Hierdurch werden Einflüsse auf einen Stream und die Bedeutung einer Qualitätssicherung deutlich. Aufbauend auf subjektiven Messverfahren werden objektive Messmethoden vorgestellt. Diese Methoden nutzen die Ergebnisse subjektiver Messungen, um ein menschlich visuelles Systems (HVS) nachzubilden. Abschließend werden Verfahren zur Dienstgütebestimmung vorgestellt, die auf objektive und subjektive Dienstegütecharakteristiken beruhen.

1 Einleitung

1.1 IP Television

Bei einer Bewertung der Dienstgüte eines Streams kann zwischen leistungsspezifischen und betrieblich allgemeinen Parametern unterschieden werden. Zur reinen Videostream Bewertung werden einzig die leistungsspezifischen Parameter herangezogen. Im Verlauf wird dargestellt, dass zu diesen technischen Komponenten allerdings noch weitere Kriterien nötig sind. Diese haben in den meisten Fällen mit dem menschlichen Wahrnehmungsvermögen zu tun.

Mit den beiden betrachteten Bereichen, der technischen und menschlichen, kann eine erste Abgrenzung der Qualitätsbewertung in die Komponenten „Quality of Service“ (QoS) und „Quality of Experience“ (QoE) vorgenommen werden.

QoS stellt hierbei eine Teilmenge der QoE da, welche durch messtechnische Mittel bestimmt werden kann. Beispielsweise ist die Laufzeit des Streams ein Faktor, der gemessen werden kann. Ein solcher Faktor hat Auswirkungen auf den Videostream und somit auch auf die Qualität. Welche Auswirkungen ein QoS Parameter hat, ist allerdings schwer zu bestimmen, da dies von vielen weiteren Faktoren abhängt. **Bild 1** und **Bild 2** stellen die Grenzen einer Qualitätseinschätzung mittels einer QoS- Messung da.

Aus Sicht einer QoS- Messung sind beide Streams mit einer identischen Fehlergröße, in diesem Fall Paketverlust, behaftet. Einem Betrachter fällt der Fehler im Stream **Bild 2** deutlicher auf, wodurch dieser Stream aus Sicht des Betrachters eine schlechtere Qualität aufweist. Dies ist somit ein Faktor, der über die reine messtechnische Messung und über eine einfache QoS Bestimmung hinausgeht. Festzuhalten ist, dass von einer QoS nicht direkt auf die vom Benutzer empfundene Qualität (QoE) zurückzuschließen ist, aber ein Zusammenhang zwischen den beiden Größen existiert. Dieser ist allerdings, wie gesehen, nicht linear.

Bild 1 Stream mit Paketverlust; Fehler in Bildmitte und über Schrift.

Bild 2 Stream mit Paketverlust; Fehler ausschließlich im Kontrastarmen Himmel

1.2 Voice over IP

Die Bewertung von Voice over IP benötigt genauso wie bei IPTV eine genauere Analyse, die über eine messtechnische QoS Bestimmung hinausgeht. So sind bei der Analyse einer Sprachqualität eine Vielzahl von Faktoren zu berücksichtigen, welche einen Einfluss auf die Ende-zu-Ende Qualität besitzen. Zu Beginn müssen die Hardwarekomponenten, wie Mikrofon und Lautsprecher, berücksichtigt werden um in weiteren Schritten den eingesetzten Codec bis hin zu dem verwendeten Netzwerk mit in die Auswertung mit aufzunehmen. Die Fehler, welche bei der Übertragung auftreten, sind typischerweise Effekte wie Echos, Verzerrungen oder Sprachpausen.

Es kann zwischen drei Arten der Analyse unterschieden werden. So ist die Betrachtung einer Hörqualität (Listening Quality), einer Konversationsqualität (Conversational Quality) oder einer Übertragungsqualität (Transmission Quality) möglich.

Genauso wie bei IPTV ist die Ermittlung einer Dienstgüte nur mittels subjektiver Verfahren möglich, welche wiederum in objektiven Verfahren eingesetzt werden, um deren Messergebnisse zu verbessern bzw. zu verifizieren.

2 Messverfahren bei IP Television

Die Definition und Gewichtung von Dienstgütecharakteristiken bei IPTV stellt ein äußerst komplexes Thema da. Die Gründe hierfür liegen in der großen Anzahl von Komponenten zur Bildkompression, -übertragung, -darstellung und dem visuellen Wahrnehmungsvermögen von Personen.

Das menschliche Wahrnehmungsvermögen nimmt hierbei eine besondere Rolle ein, da hierfür zunächst ein Verständnis entwickelt werden muss, wie ein Mensch sieht. Diesem Themenkomplex soll sich der Punkt „subjektive Messverfahren“ nähern, indem Beurteilungskriterien für subjektive Qualitäts-anforderungen gefunden und Messverfahren vorgestellt werden. Diese subjektiven Verfahren bilden im Anschluss die Basis der objektiven Messverfahren

2.1 Subjektive Messverfahren

Subjektive Messverfahren haben das Ziel, gegebene Video-Streams durch menschliche Qualitäts-einschätzungen zu bewerten. Diese Verfahren sind im Vergleich zu messtechnischen Verfahren deutlich aufwendiger, da diese erst mit einer größeren Anzahl von Probanden eine akzeptable Genauigkeit aufweisen.

Zur Klassifizierung der Qualität wurden unterschiedliche Bewertungsskalen definiert. Eine sehr häufig verwendete Skala ist der „Mean Opinion Score“ (MOS) mit einer fünf-Punkte Skala [1] wie folgende Tabelle zeigt.

Five-grade scale			
Quality		Impairment	
5	Excellent	5	Imperceptible
4	Good	4	Perceptible, but not annoying
3	Fair	3	Slightly annoying
2	Poor	2	Annoying
1	Bad	1	Very annoying

Tabelle 1 Five-grade scale

Die ITU-R hat mehrere Verfahren standardisiert, um subjektive Tests durchzuführen. Diese können in zwei Kategorien unterteilt werden. Eine Kategorie verwendet eine doppelte Anregung (double stimulus), indem ein Proband einen Qualitätsunterschied zwischen zwei Streams bewertet (Referenzvideo gegenüber gestörtem Video). Die zweite Kategorie beinhaltet Methoden, die eine Qualitätsbewertung durch einen einfachen Reiz (single stimulus) fordert. Einem Probanden wird nur ein (gestörtes) Video gezeigt. Als Beispiele eines double stimulus Verfahren sind die Methoden „double stimulus continuous quality scale“ (DSCQS) und „double stimulus comparison scale“ (DSCS) zu nennen. Eine Methode für das single stimulus Verfahren ist die „single stimulus continuous quality evaluation“ (SSCQE). Der Einsatz der jeweiligen Methode hängt von den zu bewertenden Kriterien einer Testreihe ab. So bietet z.B. das DSCQS Verfahren eine Testmethode, die sehr unempfindlich gegenüber dem Videoinhalt ist. SSCQS, als single stimulus Verfahren, besitzt diese Sensitivität nicht, bietet aber den Vorteil einer erhöhten Repräsentanz gegenüber Monitoring Messsystemen.

2.2 Objektive Messverfahren

Objektive Messverfahren bieten Methoden und Metriken mit Algorithmen, welche die Videoqualität cha-rakterisieren und eine Abschätzung, ein Mean Opinion Score, eines Betrachters liefern. Die Qualitätsschätzung kann hierbei an unterschiedlichen Punkten erfolgen. So wird bei der Data Metric ein decodiertes Video analysiert, wobei nicht auf die Videostruktur eingegangen wird. Die Picture Metric berücksichtigt diese Strukturen. So wird an dieser Stelle der verwendete Codec und dessen

Fehlereigenschaften differenziert betrachtet bzw. bewertet. Eine weitere Metrik, die Packetbased-/Bitstreambased- Metric, konzentriert sich auf Informationen die durch eine Headeranalyse zu ermitteln sind. Um an diese Informationen zu kommen, muss ein Videostream nur in gewissen Bereichen decodiert werden.

All diese Metriken fordern unterschiedliche Vergleichsinformationen, wodurch es möglich ist, diese weiter zu klassifizieren. Hierbei wird zwischen „full-reference“ (FR), „no-reference“ (NR) und „reduced-reference“ (RR) Verfahren unterschieden, welche im Verlauf dieses Kapitels genauer erläutert werden.

Um ein Verständnis der unterschiedlichen Metriken zu erhalten, werden im nachfolgenden Kapitel der Aufbau und die Übertragung eines Videostreams erläutert. Hiernach werden Punkte definiert, die eine Einteilung in die Bereiche Data-, Picture- und Packetbased-/Bitstreambased-Metric erlauben.

2.2.1 Vom Roh-Video zum Stream

Um ein Video über ein Netzwerk mit einer hohen Effizienz streamen zu können, sind einige Schritte nötig. Hierzu zählt neben der eigentlichen Codierung, mit einem effizienten Verfahren, auch ein Übertragungssystem mit geeigneten Protokollen.

In heutigen IPTV Systemen wird in aller Regel das TV Signal mittels dem H.264-Codec codiert, in MPEG-TS paketiert und über das IP Netzwerk transportiert. Aus diesem Grund wird auch an dieser Stelle die Übertragung eines Videos über das IP Protokoll anhand eines H.264 codierten Videos gezeigt.

Ein Video besteht aus einer Folge von Einzelbildern, die das menschliche Auge als Video wahrnimmt, wenn die Anzeigegeschwindigkeit der Einzelbilder mindestens 24 Bilder/Sekunde beträgt. Einem H.264 Encoder werden diese Einzelbilder blockbasiert übergeben. Neben einer reinen blockbasierten Kodierung, zur Datenreduktion, nutzt der Codec eine Bewegungskompensation und eine Intra-Frame Prädiktion.

Ein Bild kann in mehrere Slices [2] unterteilt werden, welche Regionen oder Objekte eines Bildes darstellen können. Diese Slices enthalten wiederum mehrere Macro-Blöcke, welche ihrerseits Sub-Macro-Blöcke enthalten können. Die Größen der (Sub)Macro-Blöcke sind bei H.264 dynamisch in Stufen zwischen 16x16 bis hin zu 4x4 frei wählbar. **Bild 3** zeigt eine solche Aufteilung eines Bildes in zwei Slices mit einem 16x16 Macro-Block und einem 4x4 SubMacro-Block. Die Blöcke sind dann weiter in die Bereiche luma (Y) und chroma (Cb/Cr) zu unterscheiden, welche wiederum unterschiedliche Größen beinhalten können. In diesem Beispiel ist ein Sampling von 4:2:0 bei der Macro-Blockbildung gewählt worden.

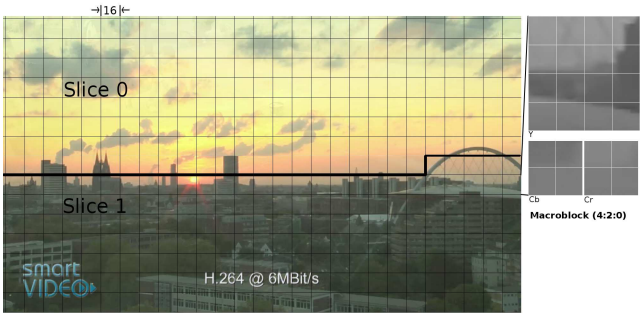

Bild 3 Bildaufteilung in Slices, Macro-, Submacroblöcke

Die nun entstandenen Blöcke werden einem H.264 Encoder übergeben, wie in **Bild 4** zu sehen ist. Nach einer Integer-Transformation und Quantisierung, in die auch die Bewegungsschätzung einfließt, entsteht ein Videobytestrom der nach einer Entropie-Codierung innerhalb eines video coding layers (VCL) [2] dargestellt wird. Dieser VCL ist netzwerkunabhängig implementiert, um die Komplexität des Codecs an dieser Stelle nicht weiter zu erhöhen. Für die Übertragung eines H.264 Streams steht der Network-Abstraction Layer (NAL) zur Verfügung. Dieser stellt ein Interface zwischen dem VCL Bytestream und der Transporteinheit bereit. Für diese Aufgabe sind neben den codierten Videodaten zusätzliche Informationen wie Sequence-Parameter-Set, Picture-Parameter-Set oder Access-Unit-Delimiter [3] nötig, welche die NAL-unit dem Stream hinzufügt.

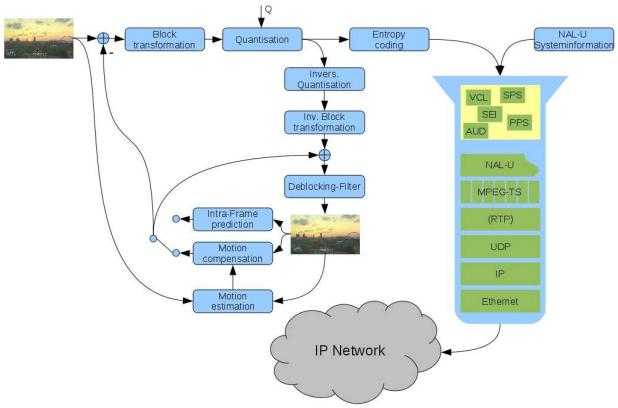

Bild 4 Video Codierung und Streaming

Eine NALU hat keine festgelegte Länge, sondern passt ihre Länge dem zu übertragenden Content an, was bei einer Übertragung über ein IP Netzwerk ungünstig ist [4]. Aus diesem Grund wird bei einem IPTV System meist ein MPEG-TS (Transport-Stream) hinzugefügt, welcher eine Paketgröße von 188 Byte festlegt. Des Weiteren bietet MPEG-TS Fehlerkorrekturmechanismen, Multiplex

Eigenschaften und Synchronisationsinformationen. Mit diesen Mitteln ausgestattet, wird ein Video Stream in UDP und IP gekapselt versendet. TCP als Übertragungsprotokoll wird nicht verwendet, da dieses Protokoll nicht für den Einsatz von zeitkritischen Diensten gedacht ist wie [5] und [6] beschreibt.

2.2.2 Metriken im Vergleich

Wie bereits erwähnt, können bei der objektiven Qualitätsanalyse unterschiedliche Bereiche eines Videostreams betrachtet werden. Je nach Ansatz hat dies gewisse Vorteile, welche sich in der Genauigkeit, Aussagekraft und der benötigter Messsystemperformance niederschlagen. Aber auch die Rahmenbedingungen unterscheiden sich, je nach Metrik, hinsichtlich der benötigten Referenzmaterialien. So werden nachfolgend die Metriken beschrieben und deren Vor-/Nachteile gezeigt. Auch wird beschrieben welche Referenzmaterialien nötig sind und unter welchen Bedingungen der Einsatz der jeweiligen Metrik sinnvoll ist.

2.2.2.1 Data Metric

Einige der am weitesten verbreiteten Verfahren zur Qualitätsmessung von Videostreams nutzen die Data Metric. Zu diesen Verfahren zählt das „peak signal-to-noise ration“ (PSNR), welchen auf der Bestimmung eines „mean squared errors“ (MSE) beruht.

Das PSNR Verfahren wird wegen der hohen Verbreitung und der relativ einfachen Berechnung gerne verwendet, zumal es auf einfachen und gut nachvollziehbaren mathematischen Verfahren beruht. Somit gehört dieses Verfahren, wie auch das MSE Verfahren, zu der Gruppe der full-reference (FR) Verfahren.

Die Berechnung des PSNR ist wie folgt definiert:

$$PSNR = 10\log_{10}\frac{\left(2^B-1\right)^2}{MSE}$$

Hierbei steht B für die Anzahl der Bit die bei einem Pixel verwendet wird, typischerweise sind dies 8-Bit und somit 256 Werte.

Der mean squared error (MSE) ist für ein Video mit der Auflösung X * Y beim Zeitpunkt T definiert als:

$$MSE_T = \frac{1}{XY}\sum_{x=1}^{x}\sum_{y=1}^{y}\left(p_T(x,y) - p_T'(x,y)\right)$$

Die Berechnung des MSE wird somit über das ganze Bild p‘ mit dem Referenzbild p berechnet.

Ein großer Vorteil des PSNR Verfahrens ist die Allgemeingültigkeit dieses Verfahrens und die große Anzahl von veröffentlichen Vergleichsmessungen.

Die Data Metric hat einige Schwächen, was eine menschliche Wahrnehmungsverzerrung und die Berücksichtigung einer inhaltlichen Wahrnehmung [7] betrifft, jedoch bietet diese eine einfache und robuste Grundlage zur Qualitätsanalyse von Videostreams. Verfeinerungen des PSNR Verfahrens, wie z.B. das locally averaged PSNR [8], bieten Möglichkeiten die Genauigkeit der Messergebnisse zu steigern und hierbei weiterhin eine Data Metric zu verwenden. Des Weiteren wurden einige Evaluierungen mit diesem Verfahren durchgeführt und ein Mapping zwischen subjektiven Tests und dem PSNR durchgeführt. Somit kann dieses Verfahren unter bestimmten Randbedingungen aussagekräftige Ergebnisse liefern.

2.2.2.2 Picture Metric

Im Gegensatz zu der Data Metric versucht die Picture Metric die Inhalte eines Bildes genauer auszuwerten. Hierfür müssen die auszuwertenden Bilder decodiert werden, was den Einsatz eines Codecs nötig macht. Auf die dann decodierten Bilder werden Verfahren angewendet, welche das menschliche visuelle System und deren Mechanismen berücksichtigt.

Bei der Picture Metric kann zwischen dem visuellen Modellansatz und einem technischen Ansatz unterschieden werden.

Der visuelle Modellansatz beruht auf einer Nachbildung des menschlichen visuellen Systems (HVS). Für die Interpretation des menschlichen Sehapparats müssen einige Faktoren, wie die Kontrast-Sensitivität oder das farbliche Sehen, berücksichtigt werden. Diese Eigenschaften wurden durch psychophysikalische Untersuchungen ermittelt.

Der technische Ansatz nähert sich der Qualitätsanalyse, im Gegensatz zum visuellen Modellansatz, weniger von der Wahrnehmungskraft des menschlichen Auges her, als dass bei diesem Verfahren die Struktur der einzelnen Codecs mit ihren Eigenschaften zum Bildaufbau im Fokus steht. Dies bedeutet allerdings nicht, dass psychophysikalische Eigenschaften keine Bedeutung bei diesen Verfahren haben. Als ein Beispiel des technischen Ansatzes ist der Structural Similarity (SSIM) Ansatz [9] von Wang zu nennen. Bei diesem Verfahren wird ein FR Ansatz verwendet. Die beiden Bilder, fehlerfrei und fehlerbehaftet, werden miteinander verglichen, wobei sich der Vergleich gleich auf drei Eigenschaften stützt. So werden die Luminanz, der Kontrast und die Struktur der

beiden Bilder miteinander verglichen. Diese drei Faktoren sind in der Bildcodierung weitestgehend voneinander unabhängig und wirken sich bei Bildfehlern auch unterschiedlich auf die Qualität aus wie **Bild 5** zeigt.

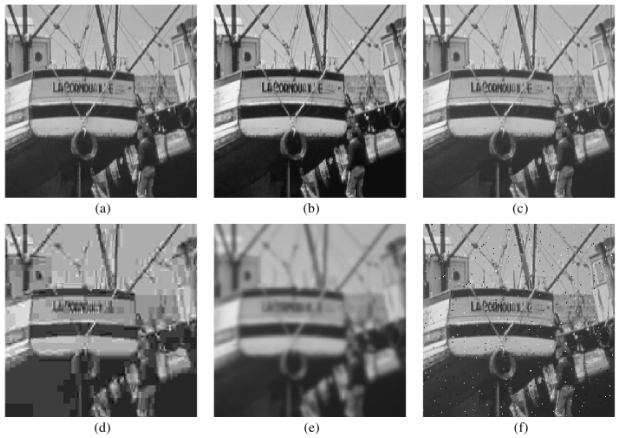

Bild 5 SSIM Bewertung - Vergleich unterschiedlich gestörte Bilder mit identischer MSE [10] a) Original b) Kontrastdehnung c) Mean-shifted d) JPEG komprimiert e) Unschärfe f) Salt-Pepper Noise

Andere Verfahren wie z.B. das in [10] beschriebene, nutzen NR Verfahren und versuchen Fehler innerhalb der Videostruktur zu ermitteln. Hierbei kann es sich unter anderem um nachfolgende Fehler [11] handeln:

- Blocking
- Blurriness
- Ringing
- Color bleeding
- Jerkiness

2.2.2.3 Packetbased-/Bitstreambased- Metric

Die zuvor beschriebenen Metriken konzentrierten sich alle auf die Analyse von Artefakten bei komprimierten Videodaten. Für den praktischen Einsatz von Messsystemen in IP Netzwerken ist diese Art der Analyse in vielen Fällen unpraktikabel. Des Weiteren stehen in diesen Fällen nicht die wie zuvor beschriebenen Bildfehlerarten im Fokus.

Bei einer Störung in einem Netzwerk entsteht aus Sicht eines Videostream ein Datenverlust, der sich als Packetloss darstellt. Dieser Datenverlust verursacht Bildstörungen, die sich allerdings nicht wie die Störungen auswirken, wie sie bei den anderen Metriken beschrieben wurden, z.B. „Contrast-stretching“ oder „salt-pepper impact“. Vielmehr drücken sich diese Fehler als Blockartefakte, Schlieren, Geisterbilder, Blockfarbfehler oder dem Gardineneffekt aus [12].

Die Packetbased-/Bitstreambased- Metric beschäftigt sich somit zentral mit der Analyse des encodierten Bitstreams. Aus diesem Grund werten die meisten Verfahren Parameter aus, die sich durch die Analyse des Bit-/Transport-Streams ermitteln lassen. Für diese Parameterermittlung muss der Videostream nicht oder nur in kleinen Teilen decodiert werden. Dies ermöglicht Messungen mit deutlich geringeren Performance-anforderungen oder die Analyse von parallelen Streams wie sie in einer typischen IPTV Plattform vorkommen. Ein Nachteil dieses Verfahrens sei jetzt schon erwähnt; die Metriken bei dieser Methode müssen für jeden Codec neu angepasst werden, da jeder Codec unterschiedliche Einflüsse bei Paketverlust verursacht. Auch ist zu beachten, dass i.d.R. bei der Bestimmung einer Qualität nicht das Video als solches bewertet wird, sondern es werden nur die Degradierungen auf das Video bestimmt und bewertet.

Zur Beschreibung dieser Metrik kann bei der Übertragung eines Videostreams, beispielsweise über eine Ethernet Verbindung, jeder Layer aus dem TCP/IP Model für sich genommen betrachtet werden. Über die Layer 1-3 können hierbei Paketverluste durch Bitfehler erkannt werden, die zu Loss führen. Aber auch weitere QoS Parameter wie z.B. Delay, Jitter oder das Burst-Silents Modell können betrachtet werden.

Innerhalb der Anwendungsschicht kann der Header des MPEG-TS Streams analysiert werden. Dies wird u.a. bei dem Standard TR 101 290 ausgenutzt, wie später noch gezeigt wird. Innerhalb des MPEG-TS Pakets wird bei heutigen IPTV Systemen i.d.R. ein H.264 Codierter Stream innerhalb MPEG4 übertragen. Die Daten eines H.264 Streams werden hierbei in Network Abstraction Layer (NAL) Units verpackt.

Neben der Betrachtung der VCL innerhalb des NAL-Unit Streams, ist die Analyse der Parameter-Sets von großer Bedeutung. Eine Bewertung von fehlerhaften Parameter-Sets wurde durch den Standard TR 101 290 definiert.

Wie bereits angedeutet, befassen sich mehrere Standards mit der objektiven Videoanalyse auf der Ebene der Packetbased-/Bitstreambased- Metric. So unter anderem TR-126 [14] des DSL-Forums und TR 101 290 [15] der ETSI, wobei sich das Dokument [15] nur mit DVB Systemen beschäftigt, dieses jedoch in der Praxis auch auf IPTV Systeme adaptiert werden kann.

Das DSL Forum unterscheidet zwischen drei Layern, dem Transport-, Application- und Service- Layer. In Zusammenhang mit dieser Metrik ist der Punkt Transport-Layer des TR-126 von großem Interesse, denn dieser Teil

beschäftigt sich, unterteilt in die zwei Bereich Control- und Data-Plane, mit den Fehlerproblematiken, die wie folgt definiert wurden:

- Zapping-Delay
- Type of Data Loss
- Loss distance
- Loss periode
- Encoding bitrate

Die nachfolgende Tabelle zeigt die durch das DSL-Forum ausgesprochenen Empfehlungen der minimalen Transport Layer Parameter für unterschiedliche Videostream Bitraten anhand eines MEPG4-AVC Streams.

Minimum Transport Layer Parameters for MPEG4-AVC						
Trans stream bit	Latency [ms]	Jitter [ms]	Max. dur. of one error	Loss Period in IP	Loss Distance [error/h]	Packet Loss Rate
1.75	<200	<50	<=16	4	1	<=6.68 E-06
2.0	<200	<50	<=16	5	1	<=7.31 E-06
2.5	<200	<50	<=16	5	1	<=5.85 E-06
3.0	<200	<50	<=16	6	1	<=5.85 E-06

Tabelle 2 Minimum Transport Layer Parameters for MPEG4-AVC [14]

Der Standard TR 101 290 [15] beschäftigt sich mit Messverfahren des MPEG Transportstroms bei DVB Systemen. Da bei IPTV i.d.R auch ein MPEG Stream zum Einsatz kommt, können einige im Standard definierte Verfahren auch bei IPTV angewendet werden.

Nachfolgende Parameter sind bei einer IPTV Messung von Interesse:

MPEG2-TS Analyse nach TR 101 290 (Auswahl für IPTV)		
Messung	Priorität	Error liegt vor, wenn
TS_sync_loss	1	min. 3 aufeinanderfolgende TS Pakete ungleich 0x47 starten.
Sync_byte_error	1	das Sync_Byte ungleich 0x47 ist.
PAT_error	1	die PAT fehlt, verschlüsselt ist, die Wiederholrate > 500ms oder die Table ID nicht 0 ist.
PMT_error	1	eine angezeigte PMT fehlt, verschlüsselt ist, die Wiederholrate > 500ms oder die Table ID nicht 2 ist.
PID_error	1	eine referenzierte PID fehlt oder die Wiederholrate >500ms ist.
Continuity_count_error	1	der continuity counter nicht dem alten continuity counter plus eins entspricht (mit Modulo 15).
PCR_error	2	die Differenz zweier PCR-Werte größer 100ms oder der Abstand zweier Pakete mit PCR-Werten größer 40ms ist.
PCR_accurancy_error	2	der PCR-Jitter, also die Toleranz zweier PCR-Werte > 500ns zueinander ist.
PTS_error	2	die Wiederholrate zweier PTS-Werte > 700ms ist.
CAT_error	2	TS Pakete verschlüsselt sind aber keine CAT übertragen wird oder die Table ID nicht 1 ist bei einer PID von 1.

Tabelle 3 MPEG2-TS Analyse nach TR 101 290 (Auswahl für IPTV) – teilweise [16]

Die Ergebnisse einer Messung nach den Richtlinien des TR-126 und TR 101 290 zeigen welche Parameter zu beobachten sind. Eine Aussage über die wirkliche augenblickliche Qualität des Streams lässt sich so allerdings nicht ausreichend genau ermitteln. Die Werte, die den beiden Standards zu Grunde liegen, geben allerdings eine Gewichtung bei der Analyse eines Videostreams über einen ungesicherten Kanal. Diese Standards stellen somit die Basis für Verfahren zur

Dienstgütebestimmung, welche eine Packetbased-/Bitstreambased- Metrik nutzen.

3. Messverfahren bei Voice over IP

Ebenso wie bei IPTV ist die Definition und Gewichtung von Dienstgütekriterien recht komplex, denn auch hier ist neben der hohen Anzahl von Komponenten das menschliche Wahrnehmungsvermögen ausschlaggebend für die Qualitätsbeurteilung.

Aus diesem Grund wird auch in diesem Kapitel beispielhaft ein subjektives [17] sowie ein objektives [18] Messverfahren vorgestellt.

3.1 Subjektive Messverfahren

Wie bereits in der Einleitung beschrieben, gibt es unterschiedliche Messverfahren, die sich mit den Bereichen Listening, Conversation und Transmission Quality beschäftigen. Bei allen Verfahren wird, wie bei IPTV, durch die Auswertung von Probanden Einschätzungen eine Qualität ermittelt. So z.B. auch bei dem absolute category rating (ACR) [19] Verfahren. Dieses Verfahren gehört zu den single stimulus Methoden.

Genauso wie bei den Verfahren von IPTV werden die Messergebnisse zu einem MOS Wert zusammengefasst. Nachstehend werden die für VoIP wichtigsten Codecs mit ihren MOS Werten gezeigt.

MOS Werte verschiedener Codecs				
Codec	Algorithm	Bit rate [kbps]	R-Factor	MOS
G.711	PCM (Pulse Code Modulation)	64	94	4,42
G.723.1	MP-MLQ (Multi-Pulse Maximum Likelihood Quantization)	6.3	79	3,99
G.723.1	ACELP (Algebraic Code-Excited Linear Predicition)	5.3	75	3,82
G.726	ADPCM (Adaptive Differential Pulse Code Modulation)	32	87	4,26
G.729	CS-ACELP (Conjugate Structure-Algebraic Code-Excited Linear Predicition)	8	84	4,17
GSM (full)	RPE-LTP (Regular Pulse Excitation Linear Prediciton)	13	74	3,78
GSM (half)	VSELP (Vector Code Excited Linear Predicition)	5.6	71	3,64

Tabelle 4 MOS Werte verschiedener Codecs [19]

3.2 Objektive Messverfahren

Ein sehr bekanntes und anerkanntes Verfahren zur objektiven VoIP Messung ist das Perceptual Evaluation of Speech Quality (PEVQ) Verfahren. Dieses ist durch die ITU-T im Standard P.862 [18] definiert worden. Es gehört zu der Gruppe der FR-Verfahren.

Für die Analyse wird das degradierte Audiosignal mit dem Referenzsignal zeitlich synchronisiert. Hiernach werden die Audiosignale verglichen und als Ergebnis wird eine MOS-ähnliche Wertung im Bereich von -0.5 bis +4.5 ausgegeben. Typischerweise liegen die Ergebnisse einer Messung im Bereich 1 bis 4.5.

Für die Bestimmung des PEVQ-Wertes werden mehrere Faktoren bestimmt, hierzu definiert [18] u.a.:

- Absolute hearing threshold
- The power scaling factor
- The loudness scaling factor
- Calculation of the pitch power densities

4. Verfahren zur Dienstgütebestimmung

4.1 IPTV

Für die Bestimmung einer Dienstgüte bei IPTV wurde eine Reihe von Verfahren entwickelt die teils sehr offen dokumentiert und diskutiert werden. Es existieren aber auch einige rein proprietäre Verfahren über deren Arbeitsweise kaum Informationen zu erhalten sind. Nachfolgend wird eine kleine Auswahl von Verfahren und Arbeitsweisen beschrieben, weitere Verfahren sind in [7] beschrieben. Die Messergebnisse der einzelnen Verfahren werden teilweise nicht als MOS dargestellt. Die Video Quality Experts Group (VQEG) beschäftigt

sich u.a. mit der Bestimmung einer Formel zur Umrechnung von Messwerten in einen MOS Wert [20].

4.1.1 PEVQ

Der Standard J.247 [23] der ITU-T beinhaltet eine Testmethode, die vom Unternehmen Opticom entwickelt wurde. Dieses Verfahren ist ein FR-Verfahren, welches im Vergleich zum PSNR Verfahren eine höhere Genauigkeit aufweist. Bei dem Pixel-zu-Pixel Vergleich werden die Eingangssignale in ihre Teile Y, Cb und Cr aufgeteilt und einzeln analysiert. Ein Bild kann hierbei in sogenannte Interessensbereiche (regions of interest - ROI) aufgeteilt werden. Die Qualitätsbestimmung wird in diesem Fall für jede ROI einzeln durchgeführt.

In einem ersten Schritt wird die wahrnehmbare Differenz bestimmt. Nach dieser Analyse von YCbCr wird eine Klassifizierung vorgenommen. Hierbei wird das Video auf mehrere Eigenschaften hin überprüft. Hierzu zählen u.a.:

- Jerkiness
- Blur
- Blockiness
- Frame Skip and Freez
- Contrast
- Brightness

(siehe hierzu auch Picture Metric)

In einem letzten Schritt, dessen Ziel die Generierung eines MOS-Wertes ist, werden die zuvor bestimmten Messwerte zusammengefasst. Als Ergänzung werden noch Parameter auf Paketebene erhoben und der MOS Generierung hinzugefügt. Hierzu zählen u.a. die Parameter Jitter, Delay und Loss.

4.1.2 V-Factor

Ein kommerzielles Verfahren zur Dienstgütebestimmung bei IPTV wurde von Symmetricom entwickelt. Dieses Verfahren analysiert die Qualität anhand mehrerer Parameter, die sich im Wesentlichen ohne ein decodieren des Bildes ermitteln lassen. Als Eingang benutzt das Verfahren den Bit- und Transportstream, hierbei benötigt das Verfahren keine Referenzen und gehört somit zu den NR-Verfahren. Hierbei werden mehrere Header des Streams und die benötigten Parameter analysiert.

Bei der Analyse des Videostreams erstellt der V-Factor einen MQUANT. Der MQUANT beschreibt die Quantisierungsgröße eines Macroblocks bei MPEG2. Dies gibt eine erste Annäherung daran, wie sich die Videokompression auf die Videoqualität auswirkt. Hierbei kann gezeigt werden, dass MQUANT eine lineare Annäherung an die Videoqualität bietet wie [24] darlegt.

Ein weiterer Bereich der zur Qualitätsanalyse herangezogen wird, ist die Komplexität des Videomaterials. Bei H.264 codierten Videos werden zur Komplexitätsbestimmung im Wesentlichen die drei Bereiche VCL Datenrate, eine Analyse der Slices mit Macroblöcke und der Verlust bestimmt. Diese Parameter werden jeweils in ein eigenes Model überführt und anschließend kombiniert, wie **Bild 5** zeigt.

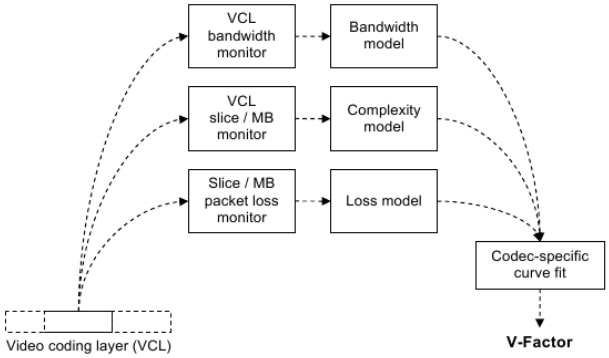

Bild 5 VCL Komplexitätsbestimmung bei H.264 für V-Factor

Das „Bandwidth model“ wird durch ein Markov Modell dargestellt und weist die drei Zustände low, average, und high auf. Zur Bildung eines „Complexity model“ wird ebenfalls ein Markov Modell verwendet, wobei jeder Macroblock einzeln analysiert wird.

Das „Loss model“ beschreibt den sichtbaren Verlust durch eine Kombination der Komplexität und des Datenverlustes.

Die Ergebnisse der einzelnen Modelle werden durch Kombination zu einem Gesamtmodell zusammengefasst. Hierbei entsteht der V-Factor, der eine angepasste Kurve zur MOS Darstellung repräsentiert.

4.1.3 TVQM

Die Telchemy Video Quality Metric (TVQM) [21] bietet eine Sammlung von Messverfahren und gibt mehrere Ergebnisgrößen einer Messung aus. Durch die Verwendung mehrerer Verfahren sollen Anhaltspunkte beschrieben werden, wo sich mögliche Fehler bei einer schlechten QoE Bewertung befinden. Hierbei unterteilt Telchemy ihre Verfahren in drei Kategorien.

- Perceptual Quality Metrics

  In dieser Gruppe befinden sich Verfahren zur Bestimmung von MOS-Werten für Video und Audio, welche auf ermittelten Daten einer Deep

Inspection beruhen. Es werden hierbei auf Daten wie Codec-Typ, Frame Rate, GoP Strukturen, Loss Informationen und Timer Informationen zurückgegriffen. Hierüber hinaus werden auch PSNR Verfahren angewendet um die Messung auch im Bereich Data/Picture Metrik zu komplettieren.

- Video Stream Metrics:

  In diesem Bereich werden Statistiken aus dem Bereich Videocodec erhoben, hierzu zählen insbesondere Daten wie I/P/B Frame Informationen.

- Transport Metrics:

  Unter dem Bereich Transport werden Loss Metriken, Forwared Error Correction Metriken (FEC), Jitter / Delay Metriken und Metriken die sich aus der TR 101 290 Definition ergeben zusammengefasst. Dies entspricht somit weitestgehend den Packetbased-/Bitstreambased Metriken.

Wie die einzelnen Metriken zusammenarbeiten und welcher Faktor innerhalb einer Metrik wie stark auf die Qualitätsbeurteilung wirkt, wird von Telchemy leider nicht öffentlich beschrieben und diskutiert.

4.1.4 SmartVideo

Dieses Messsystem unterteilt sich in die zwei Bereiche statistische Messaufbereitung und MOS Analyse. Hierfür können ein bis n Probes als Messsonden in einem Netzwerk verteilt werden. Das Messsystem analysiert hierbei alle ihn passierenden IP-Pakete. Werden IPTV Streams erkannt, so werden die jeweiligen IP Pakete einem Stream zugeordnet. Diese Art der Analyse entspricht somit voll dem NR-Verfahren mit einer Analyse von parallelen Streams. Die statistischen Parameter beziehen sich auf unterschiedlichen TCP/IP Layer und den Empfehlungen des TR 101 290, so werden folgende statistische Parameter ausgewertet:

SmartVideo Statistic Metrics		
Transport Metrics	IP, RTP, MPEG Transport Statistics	Jitter (RFC 3550) Jitter Inter-Arrival Delay (min/avg/max) Packet Loss (IP/RTP) Packet Size (Burst) Number of Bursts
	TR 101 290	Continuity count error
Video Stream Metrics		Codec type Group of Pictures Type (auto-detected) Group of Pictures Length (auto-detected) Image Size Profile/Level I/P/B frame packets received, lost, discarded (MP2) Slice Statistic (H.264)

Tabelle 5 SmartVideo Statistic Metrics

Anhand der statistischen Parameter ist schon zu entnehmen, dass das SmartVideo Messsystem auf den Packetbased-/Bitstreambased-Metriken beruht. Es werden sowohl die nach TR 101 290 als auch die nach TR-126 ausgesprochenen Empfehlungen berücksichtigt.

Der in diesem Verfahren bestimmte MOS beruht nicht auf den bereits beschriebenen Verfahren wie z.B. dem V-Factor. Vielmehr wird ein eigener Faktor bestimmt, der mittels einer Wissensdatenbank ein „Best Match“ ermöglicht. Diese Datenbank enthält unter anderem Informationen zu subjektiven Messergebnissen.

Die Komponenten, welche für eine MOS Bestimmung herangezogen werden, sind u.a. Delay, Jitter, Loss und statistische Parameter wie die Randbedingungen des physikalischen Übertragungskanals. Diese Komponenten werden zu einer Gruppe „Network-QoS“ (N) zusammengefasst. Des Weiteren ergibt sich eine zweite Gruppe, „Video-QoS“ (V), u.a. aus den Parametern Auflösung, Codec-Type und Slice-Statistik. Eine dritte Gruppe, erlaubt die Bewertung eines IPTV Streams in die Bereiche relativer und absoluter MOS zu unterteilen, um ein Verhältnis zwischen Auflösung und Anzeigesystem zu ermöglichen.

Hieraus bestimmt die nachstehende Formel einen Faktor, welcher durch die Wissensdatenbank, eine angepasste MOS Kurve erstellt und so zu einem V-MOS korreliert.

$$MOS_{SV} = a_1 \frac{1}{e^{\alpha N}} + a_2 \frac{1}{e^{\beta V}} + a_3 \frac{1}{e^{\lambda R}}$$

Die Anpassung der Kurve zur MOS Bestimmung wird hierbei durch eine subjektive Messung nach dem beschriebenem DSCQS Verfahren ermittelt.

4.2 VoIP

Auf dem Markt haben sich unterschiedliche Arten von Messgeräten zur Dienstgütebestimmung etabliert. Diese arbeiten zum Teil mit FR Verfahren, in dem sie vorgefertigte Test-Samples über ein Netz senden. Andere Verfahren versuchen eine Güte mit NR Verfahren zu ermitteln, wie z.B. das nachfolgend vorgestellte E-Modell.

4.2.1 E-Modell

Die ITU hat im Dokument G.107 [22] einige Empfehlungen hinsichtlich der Planung und Bewertung von Kommunikationsnetzen und deren Über–tragungsqualität definiert. Hieraus ist das E-Modell entstanden, welches eine objektive Sprachqualität berechnet. Als Ergebnis der E-Modell Berechnung wird der sogenannte Rating Factor (R-Faktor) ermittelt. Dieser liegt bei Verwendung von schmalbandigen Codecs im Bereich von 0-100. Grundlage der Berechnung ist die Messung des Signal zu Rauschabstands, worauf die Störeffekte linear wirken [23]. Hierdurch ergibt sich Berechnungsformel:

$$R = R_0 - I_s - I_d - I_e + A$$

R_0 ist das Verhältnis zwischen Sprachsignal und Störgeräuschen.

I_s sind die simultan wirkenden Störungen.

I_d sind die durch Verzögerungen wirkenden Störungen.

I_e sind die Auswirkungen der eingesetzten Komponenten, wie z.B. Codecs

A ist ein „Vorteils-Faktor“ der als Kompensationsfaktor eingesetzt werden kann.

Der bestimmte R-Faktor kann anschließend in einen MOS, den MOS_{CQE} umgerechnet werden. Dieser liegt wie bereits beschrieben zwischen 1 und 4,5.

$$MOS_{CQE} = 1 + 0{,}0335R + R(R-60)(100-R)7 \cdot 10$$

Durch dieses Verfahren ist es möglich, bereits während eines Gesprächs die Qualität zu bestimmen, wenn bekannte Parameter für die Berechnung bereitgehalten und Messzeitintervalle definiert werden.

5 Fazit

In diesem Bericht wurden unterschiedliche Verfahren zur Bestimmung einer Dienstgüte bei IPTV und VoIP dargestellt. Es wurde beschrieben, warum subjektive Messungen erhoben werden müssen, um objektive Messverfahren erstellen zu können. Des Weiteren wurden einzelne Messverfahren vorgestellt und eine Einordnung zu unterschiedlichen Metriken gegeben. Diese Metriken weisen bestimmte Charakteristiken auf, welche eine Untersuchung einer Dienstgüte aus unterschiedlichen Blickwinkeln erlaubt. So zeigte das PSNR Verfahren, dass eine Bewertung eines Videostreams im FR Verfahren recht einfach ist, aber auch schnell an seine Grenzen stößt. Trotzdem wird es in der Praxis gerne als erste Bewertung verwendet, da die Berechnungsgrundlage recht einfach und ressourcen-schonend bewerkstelligt werden kann.

Es wurde auch gezeigt, dass Monitorigsysteme nicht mit FR-Verfahren betrieben werden können und somit auf NR-Verfahren zurückgegriffen werden muss, welche i.d.R. ungenauer arbeiten aber dafür den Stream während des Transportes bewerten können. Diese Verfahren nutzen, wie gezeigt, als Grundlage ihrer Berechnungen Ergebnisse von SSCQE Verfahren, da diese einen NR Ansatz besser abbilden können. Aber auch DSCQS Verfahren können unter bestimmten Voraussetzungen verwendet werden.

Die in der Praxis eingesetzten Systeme nutzen meistens eine Kombination mehrerer Verfahren, um die Nachteile einzelner zu minimieren. Auch sind diese Systeme in den meisten Fällen für bestimmte Umgebungen konzipiert

Des Weiteren wurde in dieser Arbeit gezeigt, dass ebenso bei VoIP subjektive Messverfahren nötig sind, um eine Basis für objektive Messverfahren zu schaffen, welche ein HVS nachbilden. Die Verfahren sind der Analyse von IPTV sehr ähnlich, auch wenn durch die unter–schiedlichen Sinnesorgane keine Überführung der sich in den Ergebnissen wiederfindenden Erfahrungen möglich ist.

6 Literaturverzeichnis

[1] ITU-R, BT.500-12 Methodology for the subjective assessment of the quality of television pictures. Geneva, 09.2009.

[2] K.-B. Lee, "Video Coding Using the H.264/AVC Standard," in Mobile Multimedia Broadcasting

Standard: Technology and Practice.: Springer, 2009, ch. 15, pp. 435-460.

[3] International Standard, ISO/IEC 14496-10 Information technology - Coding of audio-visual objects - Part 10: Advanced Video Coding., 15.12.2005.

[4] T. Dreibholz, "Management of Layered Variable Bitrate Multimedia Stream over DiffServ with Apriori Knowledge," Uni Bonn, Bonn, Diplomarbeit 2000.

[5] S.Küffner, O.Portugall A. Grebe, "Objektives und Subjektives Qualitätsmonitoring von H.264 IP Videostreamin in Mobilfunknetzen," in Mobilkommunkation (ITG-FB 223). Osnabrück: VDE, 2010, pp. 81-86.

[6] S.Kueffner, O.Portugall A.Grebe, "Architekturkonzepte und Performance Bewertung von IPTV über Multicast in WLAN und LTE/UMTS," in ITG Mobilkommunikation (ITG-FB 215). Osnabrück: VDE, 2009, pp. 67-72.

[7] Oliver Portugall, "Objektive und subjektive Dienstgütecharakteristiken bei VoIP und IPTV," Fachhochschule Köln, www.dn.fh-koeln.de, Seminararbeit 12.2010.

[8] Snježana Rimac-Drlje, Krešimir Grgic Mario Vranješ, Locally Averaged PSNR as a Simple Objective Video Quality Metric. Zadar, Croatia, 2008.

[9] Alan Conrad Bovik, Hamid Rahim Sheikh, Eero P. Simoncelli Zhou Wang, "Image Quality Assessment: From Error Visibility to Structural Similarity," IEEE TRANSACTIONS ON IMAGE PROCESSING, vol. 13, no. 4, pp. 600-612, April 2004.

[10] S. Winkler S. Süsstrunk, "Color Image Quality on the Internet," SPIE Internet Imaging, vol. 5304, pp. 118-131, January 2004.

[11] F. Oberti J.E. Canviedes, "No-Reference Quality Metric for Decraded and Enhanced Video," in Digital Video Image Quality and Perceptual Coding.: CRC Press, 2005, ch. 10, pp. 305-324.

[12] A. Grebe, "Quellencodierung Applikationscharakteristik in Multimedia–kommunikation," FH Köln, www.dn.fh-koeln.de, 2010.

[13] R. Ravikanth R. Koodli, "RFC 3357: One-way Loss Pattern Sample Metrics," IETF - Network Working Group, 2002.

[14] DSL Forum, "Technical Report: TR-126 - Triple-play Services Quality of Experience (QoE) Requirements," Architecture & Transport Working Group, 2006.

[15] ETSI, "TR 101 290: Digital Video Broadcasting (DVB); Measurement guidelines for DVB systems," 2001.

[16] W. Fischer, Digitale Fernsehtechnik in Theorie und Praxis. München: Springer, 2006.

[17] ITU-T, "P800: Methods for subjective determination of transmission quality," 1996.

[18] ITU-T, "P.862: Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," 2001.

[19] A. Grebe, "Multimedia Services in Next Generation Networks," FH Köln, www.dn.fh-koeln.de, 2010.

[20] Video Quality Experts Group, "Validation of objective models of multimedia Qualty," 2008.

[21] Telchemy, "TVQM Video Quality Metrics," 2008.

[22] ITU, "G.107: The E-model, a computational model for use in transmission planning," 2005.

[23] T. Reichelt, "Entwicklung eines Messsystems zur Analyse von VoIP-Strömen mit R-Faktor, MOS und Spektralanalyse," FH Köln, 2007.

[24] Institute for Telecommunication Sciences (ITS), National Telecommunications and Information Administartion (NTIA), U.S. Department of Commerce, Comparing subjective video quality testing methodologies.: SPIE Video Communications and Image Processing Conference, 2003.

[25] Praveen Mohandas Stefan Winkler, "The Evolution of Video Quality Measurement: From PSNR to Hybrid Metrics," IEEE TRANS. BROADCASTING, vol. 54, no. 3, 2008.

[26] ITU-T, P 910: Subjective video quality assessment methods for multimedia applications., 2008.

[27] ITU-T, "J.247: Objective perceptual multimedia video quality measurement in the presence of a full reference," 2008.

Autor

M. Sc. Dipl.-Ing. Oliver Portugall, ist wissenschaftlicher Mitarbeiter am Institut für Nachrichtentechnik in der Forschungsgruppe Datennetze der Fachhochschule Köln.

Kontakt: oliver.portugall@fh-koeln.de

Vergleichende Analyse von Container- und Codectypen für IP-Videostreaming in IPTV-, WebTV- und Internet- Dienste

Stephan Küffner

Abstract

Im Rahmen dieser Ausarbeitung soll eine vergleichende Analyse von IP-Streaming-, IPTV- und WebTV-Diensten in All-IP-Netzen [2] unter Berücksichtigung verschiedener Container und Codectypen erarbeitet werden. Es werden verschiedene Arten von Internet-Video-Streaming Möglichkeiten in Verbindung mit typisch eingesetzten Codec- und Containervarianten untersucht. Anschließend werden die einzelnen Formate miteinander verglichen bzw. gegenübergestellt.

1 Einleitung

Aufgrund verschiedener Implementierungen von Diensten rund um das Thema Internet-Video-Streaming und der Tatsache, dass keine einheitlichen Standardisierungen existieren, führt zu einer großen Vielfalt an Codectypen und Containerformaten, die genutzt werden können. In den Bereichen IPTV und WebTV wird diese Varianz versucht in verschiedene Standards festzuhalten. Dabei gilt es zu berücksichtigen, dass jeder einzelne Codec oder Container für das Videomaterial, bzw. der Dienst auf dem das Videomaterial später ausgestrahlt wird, Vor- und Nachteile aufweist. Das DVB-Projekt beschreibt in verschiedenen Spezifikationen wie Digital Video Broadcasting (DVB) für IPTV aussehen soll. Weitere Zusammenschlüsse von verschiedenen IPTV-Anbietern finden sich im Open-IPTV-Forum, um übergreifende Spezifikationen für die Bereitstellung von Diensten für das Internetfernsehen zu entwickeln und existierende Standards zu integrieren. Im Bereich WebTV gibt es keine Projekt-Gruppe die Spezifikationen zusammenfasst, sodass der Markt an dieser Stelle stärker fragmentiert ist und Nischenlösungen entstehen. Suchmaschinen-Anbieter Google zeigt mit seiner Opensource Variante WebM, wie es funktionieren kann, ein Video mit einer guten Qualität effizient und bandbreitenschonend in einer reinen IP-Umgebung zu übertragen.

2 Internet Video Streaming

Internet-Video-Streaming ist der Oberbegriff für die Auslieferung von Videomaterial über das Internet. Das Videomaterial wird an den Internetbenutzer gestreamt (ausgeliefert), dabei wird zwischen zwei verschiedenen Streaming Verfahren differenziert

1. Live-Streaming Bereitstellung des Angebotes in Echtzeit Protokolle: RTP, RTCP, RTSP

2. On-Demand-Streaming Übertragung vom Server zum Client Wiedergabe erfolgt während Übertragung Zwischenpufferung Vor-, Zurück–spulen, Pausieren möglich Protokolle: HTTP, FTP

2.1 IPTV

Für IPTV gibt es keine spezielle Definition, die im Allgemeinen beschreibt was IPTV tatsächlich ist. IPTV wird meistens wie folgt interpretiert:

IPTV bezeichnet die Übertragung von digitalen TV-Inhalten über ein geschlossenes Netzwerk des Netzwerkproviders über das Transportprotokoll IP (Internet Protocol).

Bei IPTV werden digitale TV-Inhalte (Live-Fernsehen) einem ebenfalls geschlossenen Kundenkreis zugänglich gemacht. Dabei garantieren die ISPs den Kunden eine fest definierte Dienstgüte (QoS/QoE).

Der Empfang erfolgt über eine Set-Top-Box (STB) des Anbieters, die in der Regel an ein IAD (Integrated Access Device - DSL-Modem) angeschlossen wird. In dem Headend des Providers wird das lineare Fernsehen (Broadcast-TV) vom Satellit, terrestrisch oder per Kabelanschluss eingespeist, passend für IPTV transcodiert und anschließend in das Core-Netz des Providers mittels Multicast eingespeist. Über das Core-Netz werden die IPTV-Streams zu den Knotenpunkten im Netz verteilt. Ein IPTV-Kunde kann sich mittels seiner IAD und dem IGMP-Protokoll an der IPTV-Gruppe anmelden und teilnehmen. Der jeweilige IPTV Stream wird dann über Router, Switche und dem DSLAM bis zum Kunden freigeschaltet. Die Set-Top-Box decodiert den IPTV-Stream und zeigt das reine Videomaterial auf dem Fernsehen.

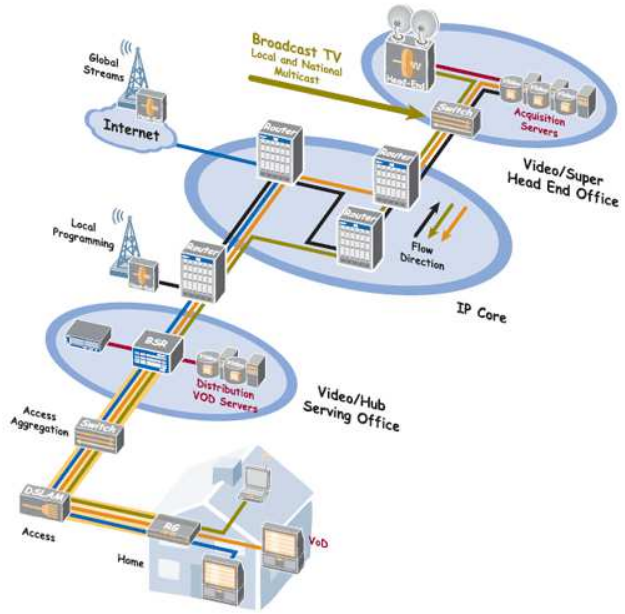

Bild 1 Typischer Aufbau IPTV-Streaming-Plattform

Ethernet-Paketrahmenaufbau und Transcodieren

Das Transcodieren der Videodaten ist erforderlich, um eine optimale Performance für das Netz in Verbindung mit IPTV zu erlangen. DVB-S, DVB-T oder DVB-C Videodaten sind nach ETSI ETR 154 (e3) [5] MPEG2 kodierte Bildinformationen die mit einem Header (MPEG-TS – min. 4 Byte) in 188 Byte große Datenpakete als Broadcast-Signal mit einer Datenrate von 2 Mbit/s bis zu 8 Mbit/s über die jeweilige Infrastruktur ausgestrahlt werden. Würden diese Informationen direkt auf die IPTV-Plattform angewandt werden, entsteht durch die nötige Kapselung in IP-Pakete ein zu großer Overhead. Die Kapselung durch die Header Ethernet (14 Byte + 4 Byte), IP (20 Byte), UDP (8 Byte) und eventuell RTP (12 Byte) würde zusätzliche 58 Byte Headerinformationen benötigen. Damit Netzwerk-Pakete effizienter genutzt werden können und die maximale Paketgröße von 1518 Byte pro Ethernetframe ohne Fragmentierung der IP-Pakete erreicht wird, werden bis zu sieben MPEG-TS Pakete in ein IP-Paket gekapselt.

Weiterhin ist die MPEG2 Kodierung nicht effizient genug für die Verbreitung von IPTV über Breitbandanschlüsse. Die Videodaten werden durch das Headend in MPEG4/H.264 transcodiert, da der Codec H.264 bzw. MPEG4 wesentlich stärker komprimieren kann [7].

2.2 Web TV

Wie auch bei IPTV gibt es bei WebTV keine einheitliche Spezifikation, bzw. einen einheitlichen Standard. Dementsprechend wird WebTV allgemein wie folgt definiert:

Beim Internetfernsehen bzw. WebTV werden Streams über das weltweit zugängliche Internet übertragen. [14]

Darüber hinaus definiert man WebTV auch unter dem Begriff „Streaming Media“, der das allgemeine Streaming zwischen einem Server und einem Client als Punkt-zu-Punkt-Verbindung entspricht. Die Datenübertragung beim Empfang von WebTV Inhalten nennt man Streaming, bzw. bei Live-Video oder Live-Audio Daten auch „Live-Streaming“.

2.3 Eigenschaften der Dienste

Bei der vergleichenden Analyse der Dienste müssen zunächst erst mal alle Eigenschaften der Dienste aufgenommen werden. Dazu werden IPTV und WebTV gegenübergestellt und zusätzlich mit Satellit, Kabel und terrestrischem Fernsehen verglichen. Die folgende Grafik zeigt „Vielfältige Möglichkeiten des modernen Fernsehens“ aus einer Studie der Bitkom.

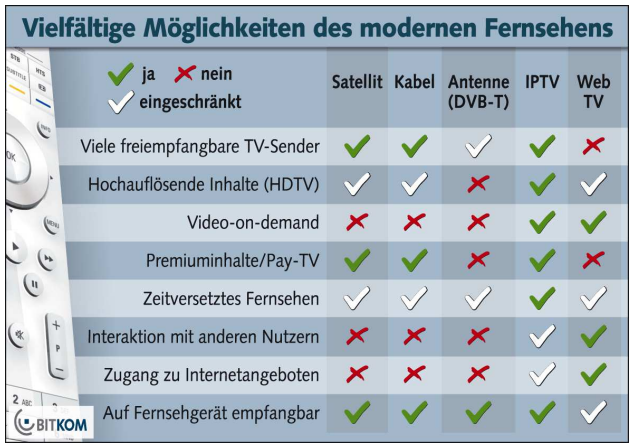

Bild 2 Fernsehen Übertragungswege im Vergleich. BITKOM [3]

Im direkten Vergleich zeigt sich ein Nachteil aller Übertragungsarten, ausgenommen IPTV und WebTV. Sie lassen keine Interaktion mit dem Benutzer zu, was aber aufgrund von linearem Fernsehen (BroadcastTV) üblich ist. Alle Dienste und Empfangsmöglichkeiten lassen ein Betrachten auf dem Fernsehen zu und bis auf WebTV ist das TV-Sender Angebot gut ausgebaut. Es zeigt sich deutlich, dass IPTV die meisten Möglichkeiten für den Benutzer bietet.

3 Container

Container dienen dazu, Video-, Audio- und Dateninhalte effizient innerhalb einer Datenstruktur (Datei o.ä.) abzulegen, damit diese von einem Abspielprogramm wiedergegeben werden können. Wie der schematische Aufbau zeigt, werden die Informationen innerhalb des Containers gruppiert, damit unterschiedliche Inhalte

(Audio, Video und Daten) in einer Datei zusammen–gefasst werden können.

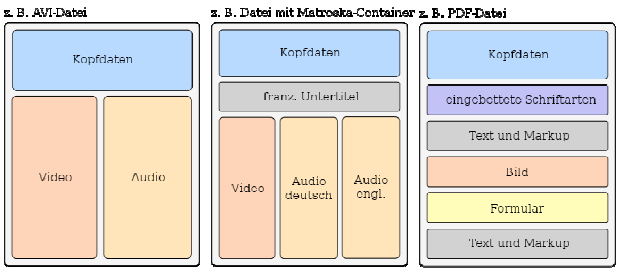

Bild 3 Schematischer Containeraufbau für Multimediadateien. WIKIPEDIA [11]

3.1 MP4

Der MP4 Container ist das Standard-Containerformat für MPEG-4-ASP/AVC-Videostreams und basiert auf dem Quicktime-Containerformat, welches von Apple 1990 entwickelt wurde. Standardisiert wurde es 2003 von der ISO/IEC unter 14496-12 und 14496-14. Die eingeführte Dateiendung ist „mp4“.

Das Dateiformat ist objektorientiert und beinhaltet verschiedene Elemente, die die Daten eindeutig identifizieren. Dadurch können innerhalb des Containers verschiedene Inhalte eingebunden sein. So wurde von der MPEG-Gruppe aus Kompatibilitätsgründen nur eine kleine Liste der möglichen Inhalte beschrieben. [15]

Art	Inhalt innerhalb des Containers
Video	MPEG-4 (Part 2, Part 10 & AVC/H.264), MPEG-2 und MPEG-1
Audio	AAC, MP3, MP2, MP1
Bilder	JPEG, PNG
Text	BIFS (z. B. Untertitel & in dieses Format umgewandeln

Tabelle 1 Fähigkeiten des MP4-Containers

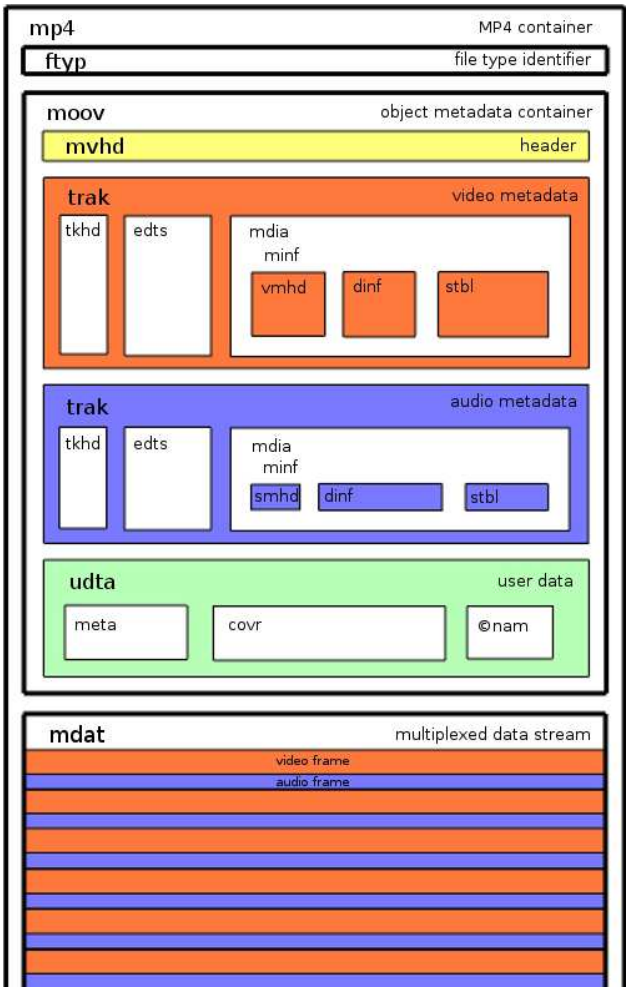

Bild 4 Beispiel Aufbau eines MP4-Container [4]

3.2 FLV / F4V

Die Bezeichnung „FLV“ steht für Flash Video von Adobe Systems. Adobe wurde 1982 gegründet und entwickelte Anfangs die Drucker Seitenbeschreibungssprache Postscript.Durch einige Firmenübernamen wurde 1997 durch ändern der Produktbezeichnung von Macromedia „FutureSplash“ hin zu „Flash 1“ der erste Flashplayer erschaffen. Endgültig benennt Adobe seinen Flashplayer, bzw. Flashprodukte innerhalb des Portfolios CS3 bis CS5 angelehnt an die Creative Suite von Adobe. Die möglichen Inhalte werden in folgender Tabelle festgehalten.

Art	Inhalt innerhalb des Containers
Video	Versionen: 7 <: Sorenson-Codec,Scr-Cap-Codec 8 <: VP6 9 <: H.264
Audio	(> Version 7),AAC (< Ver. 9)
Bilder	GIF, PNG, JPEG
Text	ActionScript Message, Normal Text

Tabelle 2 Fähigkeiten des FLV-Containers

Beim Aufbau muss zwischen zwei verschiedenen Dateistrukturen differenziert werden. Alle alten und auch aktuellen Flashangebote basieren auf dem „flv“-Dateiformat. Die parallel verfügbaren Flashinhalte bauen auf dem neuen „f4v“-Dateiformat auf, das an dem MPEG-4-AVC/H.264 Standard ISO/IEC 14496-12 angelehnt ist. [9]

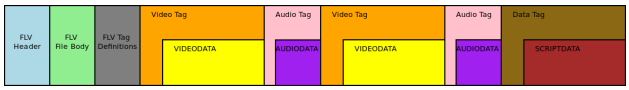

Bild 5 Beispiel Aufbau eines FLV-Containers

3.3 WebM

Seit dem von Google initiierten Projekt WebM, welches ein VP8 Videocodec mit einem Vorbis Audiocodec in einem Matroska Container beschreibt, tritt dieser als Konkurrent zum MP4 Container an. Der Container kann verschiedene Codecs für Video- und Audiodaten aufnehmen.

Das Matroska-Format ist ein binäres Containerformat. Der Ursprung rührt aus einer Meinungsverschiedenheit des Projektes MCF (Media Container Format) 2002. Im Projekt MCF wurde die Entscheidung getroffen, dass für die Struktur des Containers von der EBML (Extensible Binar Meta Language), einem binärem XML Format, zu einem einfacheren Aufbau gewechselt werden soll. Unter anderem sollte eine fixe Headergröße pro Tag eingeführt werden, damit die Hardware einen „MKV“-Container besser parsen kann. Es splittet sich 2002 die Non-Profit Organisation Matroska ab und entwickelte den „MKV“-Container mit der EBML-Struktur weiter.

Art	Inhalt innerhalb des Containers
Video	VP8, MPEG-1, MPEG-2, MPEG-4, H.264, RealVideo, WMV, Theora, Dirac
Audio	AAC, AC3, DTS, WAV, MP3, Vorbis, Flac
Subtitle	UTF-8 PlainText, ASF (Advanced Subtitle Format), Bitmaps
Buttons	MPEG/VOB PCI Packets

Tabelle 3 Fähigkeiten des WebM-Containers

Bild 6 Beispiel Aufbau eines MKV-Containers

Die EBML Struktur des Container ist simple aufgebaut, kann aber durch ihre Ähnlichkeit zu XML unzählige Tags aufnehmen. Der schematische Aufbau (**Bild 8**) zeigt nur einen kleinen Teil von dem, was sich mit dem Container realisieren lässt. Ausgenommen die Tags „Meta Seek Information“, „Cueing Data“ und „Attachment“ sind alle aufgeführten Felder zwingend erforderlich und werden benötigt um Videodaten wiederzugeben oder zu streamen. Zusätzliche Tags sind nach der Spezifikation [9] möglich, sodass zahlreiche Zusatzinformationen gespeichert werden können.

4 Codec

Der Video-Codec ist ein Algorithmus, nach dem ein Videostream encodiert und beim Betrachter wieder decodiert wird. Codecs minimieren die Datenrate eines Videos erheblich, indem sie redundante Informationen löschen. Es werden nicht alle Informationen des Einzelbildes gespeichert, sondern nur deren Differenzen zu vorherigen und nachfolgenden Bildern. Da die meisten Video-Codecs eine DCT (Diskrete Cosinus Transformation) durchführen, sind Codecs verlustbehaftete Mechanismen zur Videokompression, ähnlich wie der JPEG-Standard beim Bild. Betrachtet man ein Video, welches kodiert werden soll, wird dieses grob in 4 Schritten bearbeitet. Dazu zählen:

- Konvertierung
- Diskrete Cosinus Transformation
- Quantisierung
- Kodierung

Konvertierung

Zerlegt man das Video in seine Einzelbilder, wird am Anfang eine Farbmodell-Konvertierung durchgeführt. Im Beispiel für den H.264 Codec nach dem YCgCo-Farbmodell. Das gesamte Bild wird in dieses Helligkeit-Farbigkeit-Modell umgewandelt. Ein Farbwert wird durch eine Grundhelligkeit (Luminanz - Y) und dessen Abweichung (Chrominanz - C) von Grau in Richtung Grün (Cg) und Orange (Co) bestimmt.

Die meiste Information liegt in der Grundhelligkeit, sodass man nur noch die Abweichungen nach Grün und Orange darzustellen braucht.

Bild 7 YCgCo-Farbmodel.

Diskrete Cosinus Transformation

Eine Bild wird bei der Komprimierung mit variablen Blöcken zu je 4x4 bis hin zu 16x16 Pixeln gerastert. Ein solcher Block wird als Vektor, bestehend aus Pixelwerten (den Koeffizienten) eines geeigneten Vektorraums interpretiert. Die Koeffizienten werden mit einer vordefinierten Basis (abhängig unter anderem vom Codec) gewichtet. Um an die Koeffizienten zu gelangen, werden alle Blöcke mit Hilfe der nachstehenden Formel berechnet.

$$d_{mn} = \frac{1}{4} c_m c_n \sum_{i=0}^{7} \sum_{j=0}^{7} a_{ij} \cos \frac{(2i+1)m\pi}{16} \cos \frac{(2j+1)n\pi}{16}$$

Bild 8 Allgemeine DCT-Formel zur Berechnung von Koeffizienten

Dabei werden die Pixelwerte in Frequenzbereiche umgesetzt. Große regelmäßige Flächen im Bild schlagen sich in niedrigen Frequenzanteilen nieder, feine Details und genaue Auflösung von Farbunterschieden in hohen Anteilen. Die DCT nutzt die Schwächen des menschlichen Auges und filtert die hohen Ortsfrequenzen heraus, die das Auge ohnehin nicht wahrnehmen kann. Da sich benachbarte Pixelwerte in der Regel kaum unterscheiden, werden nach der DCT nur der DC-Koeffizient und einige niederfrequente AC-Koeffizienten signifikante Werte haben. Die anderen Koeffizienten werden annähernd Null oder gleich Null sein und können bei der Übertragung weggelassen werden.

Quantisierung

Zusätzlich werden die Koeffizienten noch quantisiert, das bedeutet, dass alle Koeffizienten durch eine Ganzzahl dividiert werden. Durch die Quantisierung wird die Genauigkeit der DCT beeinflusst. Bei einer großen Ganzzahl als Dividend werden noch mehr Koeffizienten Null. Es wird eine stärkere Komprimierung erreicht, jedoch zum Nachteil der Bildqualität.

Kodierung

Bei der Kodierung werden Codec-spezifische Verfahren angewandt, die in den nächsten Abschnitten beschrieben werden. Es werden die, aus Sicht eines Dienstes wie IPTV oder WebTV, wichtigsten Codecs analysiert.

4.1 H.264

H.264 ist der führende Industrie-Standard für Videokompression und speichert Videos in einem Format ab, welches einen geringen Speicherverbrauch bei der Übertragung aufweist. H.264 wurde ebenfalls von der MPEG-Gruppe entwickelt und ist für Geräte mit einer schmalen Bandbreite und schwächere CPUs gedacht (u.a. Handys). Der Codec schafft eine bis zu 50% bessere Komprimierung gegenüber seinem Vorgänger MPEG-2 [3]. Das wird erreicht durch eine dynamische Codierung der Bilddaten und mehr Flexibilität bei der Blockgenerierung.

Der Kodierprozess bei einem H.264 Encoder setzt sich aus der obengenannten Konvertierung, Transformation (DCT) und der Quantisierung zusammen. Zusätzlich wird nach der Konvertierung in das Farbmodell noch eine „Prediction“ vom Encoder vorgenommen. Dazu werden die Einzelbilder in 16x16 Pixel große Makroblöcke aufgeteilt und mit schon kodierten Daten aus dem gleichen Frame (Intra Prediction) oder auch kodierten Daten aus vorherigen Frames (Inter Prediction) korreliert. Bei Intra Prediction sind noch Blockgrößen von 4x4 möglich. Bei Inter Prediction können alle Varianten von 16x16 bis runter zu 4x4 Pixeln benutzt werden. Im nächsten Schritt werden nur noch die berechneten Prognosen (Motion Vectors und References) übertragen und nicht die Rohdaten. Diese Art von berechneten Daten wird in speziellen Slices (Frames) abgespeichert. So werden original Daten in I-Slices (Inter-Slice), Referenzen in P-Slices (Predicted-Slices) und die Bewegungs-Vektoren in B-Slices (Bidirectional-

Predicted-Slices) gespeichert. Nach der Quantisierung wird das Bitstream encodieren vorgenommen. Dazu werden die aufgelisteten Parameter in syntaktischen Elementen innerhalb des Binärcodes mit Hilfe von „Variable-length Code“ und „Arithmetic Coding“ kompakt gespeichert.

4.2 VP8

VP8 wurde von On2 Technologies entwickelt und von Google gekauft. Danach stellte Google den Codec unter eine Open-Source Lizenz. Der Codec kann frei verfügbar überall eingesetzt werden. Genauer spezifiziert wird er unter dem, von Google initiierten Projekt WebM, das aus technischer Sicht ein Video-Containerformat ist, welches als Container auf Matroska basiert und den Audiocodec Vorbis enthält. Dem Projekt WebM wird in Aussicht gestellt, der Standard für HTML5 Video zu werden.

Der VP8 Codec ähnelt stark dem Encoder Verhalten des H.264 Codec, jedoch sind im Detail einige Unterschiede, die den Codec nicht ganz so effektiv aussehen lassen. Der Codec unterstützt ausschließlich des Farbmodell YUV im 8-Bit Profil 4:2:0. Dieses Farbmodell unterscheidet sich von dem YCgCo- oder YCrCb-Farbmodell nur geringfügig in der Skalierung und ist in der Berechnung aufwendiger. Der Codec besitzt zudem noch inoffiziell die Unterstützung für das Farbmodell RGB. Für die Aufteilung des Bildes in Makroblöcke der Größe 16x16 Pixel werden zusätzlich noch Sub-Makroblöcke der Größe 4x4 gebildet. Das führt dazu, dass die Motion Prediction, wie es beim H.264 Codec berechnet wird, genauer ist. Allerdings können diese Informationen nicht mehr in P- und B-Slices gespeichert werden, sondern nur noch in P-Slices, da B-Slices nicht unterstützt werden. Weiterhin wird bei der Prediction nur maximal auf 3 referenzierte Frames zurückgegriffen, deren Informationen in P-Slices gespeichert werden. Sind die Frames aufgeteilt und konvertiert wird auch bei dem VP8-Codec eine DCT durchgeführt. In einzelnen Profilen bzw. Einstellungen kann hier sogar auf eine WHT (Walsh-Hadamard Transformation) zurückgegriffen werden, die zwar komplexer jedoch auch präziser ist. Bei der Speicherung in dem Bitstream als kodiertes Video wird ebenfalls eine „Arithmetic Code“-Implementierung angewandt. Der Bitstream ist in verschiedene Teile (Blöcke) unterteilt. Ein Block von komprimierten Videodaten wird in einer „Partition“ zusammengefasst. Vor der jeweiligen Partition sitzt ein Header mit Frameinformationen. Weiterhin werden in einer möglichen zweiten Partition Meta- bzw. Dekodier-informationen angehangen. Die Größe eines Blocks ist dabei variabel.

4.3 VP6

Auch von der Software Schmiede On2 Technologies entwickelt, existiert der Videocodec VP6. Dieser ist ein proprietärer Video-Codec um verlustbehaftet Video-Daten zu komprimieren. VP6 ist zur persönlichen, unkommerziellen Nutzung seit 2003 freigegeben und wird von der freien Encoder und Decoder Bibliothek „libavcodec“ unterstützt. Im August 2005 wurde VP6 von Macromedia (mittlerweile Adobe) als neues Standard-Videoformat für den Flash-Player in der Version 8 bzw. das Flash-Video-Containerformat (FLV) ausgewählt.

Der Codec schnitt 2004 unter idealen Bedingungen der Encodierung eines Videos besser ab als das gleiche Video kodiert mit dem H.264 Codec. Wie in dem folgenden Bild zu erkennen ist, existieren beim H.264 kodierten Testvideo mehr Artefakte als bei dem VP6 Codec.

Nach der Farbmodell-Konvertierung wird das Bild in 16x16 Pixel große Makroblöcke aufgeteilt. Anschließend werden die Blöcke gekennzeichnet, ob diese Bewegungsreferenzen oder Bildinformationen enthalten. Bei Bewegungsreferenzen werden diese später in P-Frames gespeichert und Bildinformationen in I-Frames. Die aus H.264 bzw. MPEG-Codecs bekannten B-Frames werden bei dem VP6 Codec nicht unterstützt. Nach der DCT werden die Koeffizienten mit Hilfe der Entropy Kodierung im Bitstream gespeichert. Dazu wird auf das 16-bit „Range Coding“-Schema zurückgegriffen. Die kodierte Video-Datei besteht dann aus binären Paketen variabler Länge mit einem Header für grundlegende Informationen zum Dekodieren des Videos.

Der VP6 Codec existiert in verschiedenen Varianten, deren Unterschied in der Qualität des Ausgangsmaterials liegt, ähnlich dem H.264 Codec:

- VP6 codec, VP60 (Simple Profile)
- VP61 (Advanced Profile)
- VP62 (Heightened Sharpness Profile)

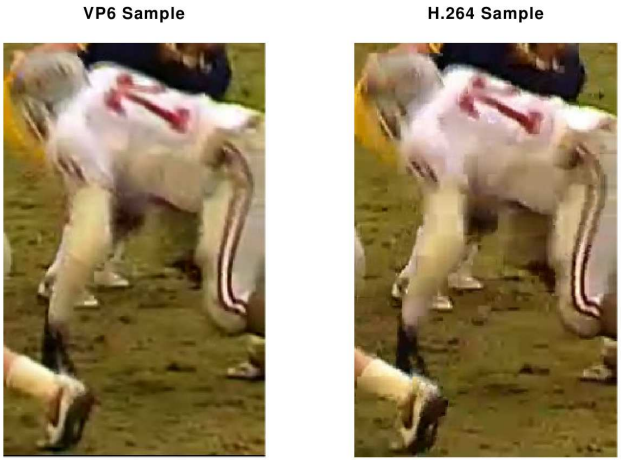

Bild 9 Codec-Vergleich zwischen VP6 und H.264.

4.4 Theora

Theora stammt wie die letzten beiden Codecs auch ursprünglich von On2 Technologies. Dort lief er unter der Bezeichnung VP3 (Genauer: VP3.1). Diese haben 2002 den Code lizenzfrei herausgegeben. Auf dieser Basis wurde dann 2004 Theora entwickelt. Theora enthält alle Features des VP3.1 Codec zum Entwicklungsende 2004. Die beiden Codec sind zwar kompatibel zueinander, jedoch kann ein Video nicht verlustfrei von Theora nach VP3 konvertiert werden.[6]

Wie die meisten Codec benötigt auch Theora als Eingangsmaterial Videos mit Vollbildern, jedoch mit einer konstanten Datenrate. Am Anfang wird das Einzelbild in ein Farbmodell konvertiert. Dieses ist für Theora das Farbmodell YCbCr und unterstützt im Gegensatz zu VP8 mehrere Farbprofile in der maximalen Auflösung von 8-Bit. Nach der Konvertierung erfolgt eine Blockgenerierung des gesamten Bildes für die DCT in der Größe von maximal 8x8 Pixeln. Das entspricht den Codecs aus der Reihe MPEG-1, -2 und H.263. Sind die Koeffizienten errechnet, so unterscheiden sich die Codecs jedoch essentiell. Wie auch bei dem VP8 Codec gibt es ausschließlich nur I-Frames und P-Frames. Es werden keine B-Frames unterstützt oder Alternativen angeboten, sodass die Kompressionsstärke nicht aktuell zu vergleichen ist. Werden die Frames kodiert, so wird am Anfang ein Reference-Frame (I-Frame) gebildet, auf das sich die weiteren Frames (P-Frames) dann beziehen können, bis zum nächsten Auftreten eines Reference-Frame, deren Auftreten variabel ist. Für die Speicherung im Bitstream werden vom Encoder alle Frames angenommen, komprimiert und anschließend in ein „Roh“-Paket inklusive Header abgelegt. Vom Codec selber werden keine Fehlerkorrekturmechanismen implementiert. Im Header selber befinden sich Zeitstempel zum Synchronisieren wieder, damit der Decoder diese Pakete in Reihenfolge annimmt, mit Hilfe der Zeitstempel in die korrekte Reihenfolge bringt und abspielt.

5 Vergleichende Analyse

Bei der vergleichenden Analyse werden die einzelnen Container und Codectypen miteinander verglichen. Wichtig ist hier zu erwähnen, dass alle Container und auch Codecs einen unterschiedlichen Verbreitungsgrad haben. So weist der H.264 Codec die meisten Fähigkeiten auf, da dieser standardisiert, sehr effizient und performant ist und auch Hardwarehersteller anhand dieser Spezifikation Geräte (z.B. Set-Top-Box) entwickeln können, womit sich Dienste (z.B. IPTV) realisieren lassen. Flash ist durch YouTube sehr bekannt geworden, jedoch durch das proprietäre Format immer an das Softwarehaus Adobe gebunden.

Speziell bei IPTV-Übertragung ist es üblich, H.264 kodierte Daten als Standard Videomaterial zu verwenden. Wie im Kapitel 3.2.1 beschrieben, wird der ganze Container in NAL Units umgepackt und in einem MPEG-TS (Transportstream) gekapselt. Anschließend werden die Pakete über das Netzwerk verschickt. [12]

5.1 Die Fähigkeiten der Container im Vergleich

Bei dem Vergleich der einzelnen Containerformate kommt es auf deren Fähigkeiten für verschiedene Einsatzbereiche an.

Art	Lizenz	Browser	IPTV	WebTV
MP4	Ja	Ja	Ja	Ja
FLV	Ja	ja	Nein	Ja
WebM[10]	Nein	Ja/Nein	Nein	Ja/Nein
OGG[8]	Nein	Ja	Nein	Ja
3GP[1]	Ja	Nein	Nein	Nein

Tabelle 4 Container-Vergleich

Durch die weite Verbreitung des MP4 Containers und der zugrundeliegenden ISO/IEC Standardisierung besitzt der Container die meisten Fähigkeiten, da er noch zusätzlich zum Flash FLV/F4V IPTV Unterstützung hat. Der Flashcontainer ist am Weitesten verbreitet, jedoch muss zum Anzeigen bzw. Abspielen des Videos das Browserplugin von Adobe System Inc. installiert sein. Das Matroska/WebM und OGG-Format sind die einzigen lizenzfreien, bzw. unter einer freien Lizenz stehenden

Containerformate. Beide Formate sind abspielbar in dem Browser, wenn man ebenfalls ein Zusatzplugin installiert. Das für mobile Endgeräte entwickelte Format 3GP ist aufgrund der MPEG-4 Spezifikation lizensiert. Die meisten Handys nutzen dieses Format für Ihre Handykameras. Prinzipiell wurde das Format nicht zum Streamen ausgelegt, ist aber mit einem speziellen Streamingserver (z.B. Helix DNA Server) möglich.

Liefert ein Streamingserver TV-Material aus, so ist mittels Browserplugin auch WebTV, wie zum Beispiel Zattoo, möglich.

5.2 Die Fähigkeiten der Codecs im Vergleich

Bei dem Vergleich der einzelnen Codectypen ist im Bereich Browser und IPTV-Fähigkeit ein ähnliches Bild zu sehen wie es bei den Containern der Fall ist. Für IPTV wird aufgrund der zahlreichen Codec-Implementierung in Geräten auf H.264 zurückgegriffen, genauso wie es auch ab Flash Version 8 im F4V-Container der Fall ist. Der VP6 Codec leistet die gleichen Dienste wie der VP8 Codec, jedoch ist der VP8 flexibler als VP6 und nicht an Adobe gebunden. Zusätzlich könnte er durch das offene Projekt WebM zum Standard für HML5 Videodienste werden. Theora als leichter Außenseiter ist der älteste Codec im Feld, dafür aber auch unter einer freien Lizenz und streamingfähig mit Hilfe des OGG-Containers. Nachteil des Codec ist die fehlende Fehlerkorrektur, welche ihn noch anfällig für Bildstörungen im Falle eines Paketverlusts macht. Beim Abspielen des Videos entstehen dann Artefakte und Bildaussetzer, die das Betrachten erschweren und das subjektive Empfinden des Benutzers mindern.

Art	Lizenz	Browser	IPTV	Stream
H.264	Ja	Ja	Ja	Ja
VP8	Ja	ja	Nein	Ja
VP6 [13]	Ja	Ja	Nein	Ja
Theora [6]	Nein	Ja	Nein	Ja

Tabelle 5 Codec-Vergleich

6 Fazit

Alle modernen und aktuell vorgestellten Container und Codectypen unterstützen Streaming über das Internet, bzw. einem Netzwerk. Der Unterschied liegt, wie die Analyse gezeigt hat, im Detail. So existieren in manchen Konstellationen keine Fehlerkorrekturen, wenn Daten bei der Übertragung verloren gehen. Der H.264 Codec in dem MP4 Container ist der Defacto-Standard in der Industrie und sehr weit verbreitet. Er ermöglicht eine sehr hohe Flexibilität was das Abspielen von Videos auf verschiedenen Endgeräten betrifft. Auch ist er bestens dafür geeignet, dank der Standardisierung, Videos effizient über das Netzwerk zu übertragen. Weiterhin wurde in der Analyse deutlich, dass der VP8 Codec in Googles WebM Projekt ein starker Konkurrent zum lizenzbehafteten H.264 Codec ist. Dieser kann sich in absehbarer Zeit zum Standard für HTML5 Embedded Videos etablieren, da er ein fast identisches Ausgangsmaterial liefert bei gleicher Vielfalt an Einstellmöglichkeit zum Komprimieren des Videos. Allgemein wurde bei der Analyse festgestellt, dass bei WebTV und Browserdiensten mit Videoinhalt immer ein Plugin installiert sein muss, damit das Dekodieren in der Anwendung funktioniert. Ein portieren eines Flash oder VP8 Codecs auf eine Set-Top-Box ließe theoretisch auch IPTV/WebTV mit den beiden Codec zu, damit man Live-TV sehen kann, ohne Nutzung des H.264 Codecs. [12]

7 Literatur

[1] 3GPP. TS 26.244 Transparent end-to-end packet switched streaming service (PSS) (Rel.6). 3GPP 3rd Generation Partnership Project, March 2005.

[2] 3GPP. TS 22.978 All-IP network (AIPN) fensibility study (Rel.9). 3GPP 3rd Generation Partnership Project, December 2009.

[3] BitKom. Internetfernsehen boomt. http://www.bitkom.org/files/documents/BITKOM-Presseinfo_IPTV_17_06_2009.pdf (2009).

[4] K. Brandenberg. Aac audio and the mp4 media format.http://www.jiscdigitalmedia.ac.uk/audio/advice/aac-audio-and-the-mp4-media-format/ (02.2010).

[5] ETSI. TR-154 Implementation guidelines for the use of and terrestrial broadcasting applications. Digital Video Broadcasting (DVB), 1997.

[6] X. Foundation. Theora speci_cation. http://www.theora.org/doc/Theora.pdf (09.2003).

[7] Grebe, Portugall, Kueffner. Objektives und subjektives Qualitätsmonitoring von H.264 IP Videostreams in Mobilfunknetzen. Mobil–kommunikation- VDE Verlag, (216), Mai 2010.

[8] IETF. RFC 3533. http://tools.ietf.org/html/rfc3533 (05.2003).

[9] A. S. Inc. Adobe Flash Video File Speci_cationVersion 10.1. Adobe Systems Inc, August 2010.

[10] Matroska. Technical speci_cations. http://www.matroska.org/technical (2010).

[11] E. Rohlicek. Containerformate. http://de.wikipedia.org/wiki/Containerformat (07.2006).

[12] Stephan Küffner. Fullpaper – Vergleichende Analyse von Container- und Codectypen für IP-Videostreaming in IPTv-, WebTv- und Internetdienste. Januar 2011.

[13] O. Technologies. VP6 Advantages White Paper, 2004.

[14] Wikipedia. Wikipedia - Internetfernsehen. http://de.wikipedia.org/wiki/ Internetfernsehen (11.2010).

[15] Wikipedia. Wikipedia - mp4. http://de.wikipedia.org/wiki/MP4 (08.2010).

[16] Wikipedia. YCgCo-Farbmodell. http://de.wikipedia.org/wiki/YCgCo-Farbmodell (2010).

Autor

M. Sc. Dipl.-Ing. Stephan Küffner, ist wissenschaftlicher Mitarbeiter am Institut für Nachrichtentechnik in der Forschungsgruppe Datennetze der Fachhochschule Köln.

Kontakt: stephan.kueffner@fh-koeln.de

Extraktion und Analyse von VoIP-Daten zur Identifikation von Telefon-SPAM

Bernhard Mainka

Abstract

Mit Telefon-SPAM werden unerwünschte Werbeanrufe mit wiederholt eingespieltem Audiomaterial bezeichnet. Zur automatisierten Identifikation und Abwehr von Telefon-SPAM in VoIP Netzwerken werden Mechanismen zur Datenextraktion benötigt. Die offensichtliche Möglichkeit, diese Extraktion durch ein aktiv an der Telefonverbindung beteiligtes System vorzunehmen, ist nicht immer praktikabel. Weitgehende Unabhängigkeit bietet die Extraktion durch Analyse einer Kopie der Netzwerkdaten, worauf der Fokus dieses Beitrags liegt.

1 Einführung

Mit SPAM over Internet Telefonie, kurz SPIT, werden in VoIP-Netzen unerwünschte Werbeanrufe bezeichnet. Begünstigt durch steigende Teilnehmerzahlen, geringe Verbindungskosten und die Möglichkeit mit wenig Aufwand automatisiert viele parallele Verbindungen aufzubauen, könnte SPIT ähnlich problematisch werden wie Email-SPAM. Im Gegensatz zur Email-Kommunikation handelt es sich bei der Telefonie um ein synchrones Medium, dessen Echtzeitcharakteristik den Benutzer zum sofortigen Handeln, also zur Gesprächsannahme auffordert.

Aktuelle Forschungsaktivitäten beschäftigen sich mit der Entwicklung von SPIT-Filtern, die möglichst effektiv vor Anrufen mit wiedereingespieltem Audiomaterial schützen. Dazu kann eine Analyse der Signalisierungsdaten durchgeführt werden. Außerdem ist es möglich die Audiodaten zu analysieren. Die dazu notwendigen Extraktionsmechanismen unterscheiden sich in VoIP-Netzen prinzipiell von denen der herkömmlichen Telefonie.

Dieser Artikel behandelt zunächst die Signalisierung und den Medientransport in Telefonnetzen. Danach werden verschiedene Möglichkeiten der SPIT-Identifikation und -Abwehr dargestellt. Anschließend werden einige relevante Aspekte des Session Initiation Protocol (SIP) [1] besprochen und die Verfahren der Datenextraktion, insbesondere die Extraktion an einem Monitoring-Port, näher erläutert.

2 Grundlagen

Unabhängig von der zugrundeliegenden Technik des Telefonnetzes unterscheidet man zwischen der Signalisierung (Verbindungssteuerung) und den Mediendaten (Nutzdaten).

2.1 Signalisierung

Ein Signalisierungsprotokoll dient dem Auf- und Abbau von Kommunikationsbeziehungen, sowie der Bereitstellung und Konfiguration von Netzwerkressourcen. In den bisherigen analogen Telefonnetzen wird sogenannte Innenband-Signalisierung eingesetzt. Das dabei verwendete Mehrfrequenzwahlverfahren nutzt den Sprachkanal als Übertragungsmedium. Außenband-signalisierung findet im digitalen Telefonnetz sowie bei VoIP Verwendung. Ein ISDN BRI (Basic Rate Interface) beispielsweise teilt den Übertragungskanal in einen D-Kanal (Signalisierung, Protokoll DSS1) und zwei B-Kanäle (Nutzdaten) auf.

In IP-Netzen sind H.323 und SIP bekannte Signalisierungsprotokolle. Mit ihnen werden logische Verbindungen zwischen zwei Kommunikationspartnern aufgebaut und außerdem virtuelle Kanäle zur Übertragung der Mediendaten konfiguriert. SIP ist heute das bevorzugte Protokoll zur Session Steuerung. In modernen NGN- und NGI-Netzwerken nimmt SIP eine zentrale Rolle ein.

2.2 Mediendaten

Anders als in leitungsvermittelten Telefonnetzen, in denen zur Medienübertragung feste Ressourcen zugewiesen werden, handelt es sich bei IP-Netzen um ein gemeinsam genutztes Medium. Die eigentlichen Nutzdaten (Audio, Video) werden mittels Real-Time Transport Protocol (RTP) übertragen, das auf dem verbindungslosen und mit wenig Overhead belasteten UDP aufsetzt. RTP bietet zusätzlich zu UDP Zeitstempel, Sequenznummern und eindeutig den Gesprächsteilnehmern zugeordneten Identifikations–nummern.

2.3 Anwendungen für die Datenextraktion

Beispiele von Anwendungen, die Signalisierungs- und oft auch Mediendaten aus VoIP-Netzen verarbeiten, sind

- Quality of Service Monitoring
- Call Center Monitoring
- vertragsrechtlich relevante Absprachen zwischen Geschäftspartnern
- Telekommunikationsüberwachung (Lawful Interception)
- Erkennung von SPIT durch Analyse der Signalisierung
- Erkennung von SPIT durch Analyse der Mediendaten (Projekt VIAT [2])

Einige Anwendungen sind schon aus leitungsvermittelten Telefonnetzen bekannt. So zum Beispiel die Tele–kommunikationsüberwachung durch staatliche Ein–richtungen. Grundlegend geändert hat sich jedoch die Datenbeschaffung, auf die in Abschnitt 5 näher eingegangen wird.

Im Projekt VIAT wird ein neues Verfahren zur SPIT Erkennung entwickelt. Die Besonderheit liegt darin, dass dabei die Audiodaten analysiert werden, was in Abschnitt 3.3 näher erläutert wird.

3 Identifikation von SPIT

Im Folgenden wird auf verschiedene Möglichkeiten eingegangen, SPIT zu identifizieren und zu blockieren. [3]

3.1 Benutzerrückmeldung

Der Angerufene hat die Möglichkeit mittels Wahlziffern (bisher bekannt als DTMF) SPIT zu melden. Überschreitet ein Anrufer einen bestimmten Schwellwert an negativen Bewertungen, könnten weitere Ver-bindungsaufbauten dieses Anrufers unterbunden werden.

3.2 Analyse der Signalisierungsdaten

Bei der Analyse der Signalisierungsdaten wird auf bestimmte Verhaltensmuster getestet. Dazu können u.a. folgende Informationen verwendet werden:

- Verbindungsversuche pro Zeit
- parallele Verbindungen
- durchschnittliche Länge der Verbindungen
- Verbindungen beendet durch Angerufenen
- hohe kumulative Verbindungsdauer
- fehlgeschlagene Verbindungsversuche
- Endgeräte-Identifikation

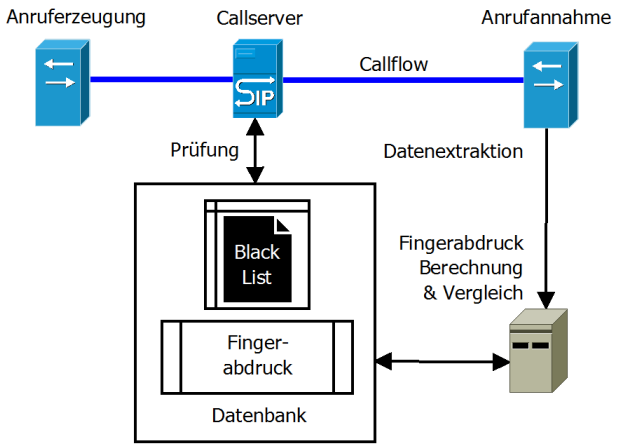

Bild 1 schematische Darstellung des VIAT Prototyps

Bei unüblichen Verhaltensmustern, also bei Überschreitung definierter Grenzwerte oder Kombinationen von Grenzwerten, wird die Anrufer-Identität als auffällig gekennzeichnet. Danach sind weitere Schritte denkbar, beispielsweise die Aufnahme in eine Blacklist oder eine manuelle Untersuchung der gesammelten Informationen.

3.3 Analyse der Audiodaten

Bei der Analyse der Audiodaten geht es darum diese auf Gleichheit oder Ähnlichkeit zu prüfen. Dies gestaltet sich insgesamt aufwändiger. Die zu verarbeitende Datenmenge ist um ein Vielfaches höher, als bei der reinen Signalisierungsanalyse. Das Ergebnis ist jedoch vielversprechend, denn theoretisch kann nach zwei Verbindungen mit den gleichen Audiodaten bereits der dritte Verbindungsversuch abgewiesen werden.

Die technische Herausforderung dieses Verfahrens liegt beim effizienten Vergleich der Audiodaten, mit zusätzlicher Robustheit gegenüber Änderungen, beispielsweise durch unterschiedliche Audiocodecs. Diese Anforderung wird im Projekt VIAT untersucht. Es existiert ein Prototyp, der in **Bild 1** schematisch dargestellt ist. Die Systeme "Anruferzeugung" und "Anrufannahme" dienen der Simulation von Anrufszenarien. Letzteres extrahiert die Signalisierung und die Audiodaten des Anrufers, wobei lediglich die ersten Sekunden zur Erzeugung eines eindeutigen Fingerabdrucks genutzt werden. Dieser wird in einer Datenbank gespeichert und mit allen bereits vorhandenen verglichen. Bei ausreichender Ähnlichkeit (Schwellwert) kann davon ausgegangen werden, dass es sich um wiedereingespieltes Audiomaterial handelt. In diesem Fall wird die Adresse des Anrufers in eine Blacklist eingetragen, so dass der Callserver (SIP Proxy) zukünftige Verbindungsversuche dieses Anrufers ablehnen kann.

4 Session Initiation Protocol

Bei SIP handelt es sich um ein sehr flexibles, an HTTP angelehntes Protokoll zum Auf- und Abbau von Kommunikationsbeziehungen. Die aktuelle Version 2.0 wurde im Jahre 2002 von der Internet Engineering Task Force (IETF) als RFC 3261 standardisiert. Es existieren weitere RFCs, die diesen Standard erweitern [1].

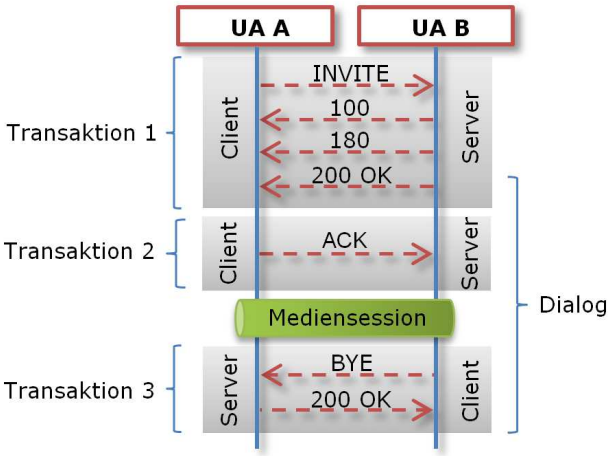

Bild 2 exemplarischer SIP Verbindungsauf- und Abbau

SIP Nachrichten werden durch ein Protokoll der Transportschicht übertragen. Häufig wird verbindungsloses UDP verwendet. Verbindungs–orientiertes TCP ist möglich, jedoch bietet SIP eigene Mechanismen an, um sicherzustellen, dass alle Nachrichten beim Empfänger angekommen sind: 3-Wege Handshake, Zeitsteuerung und Paketwiederholung

Dazu werden Nachrichten zu Transaktionen zusammengefasst (**Bild 2**). Eine Transkation wird durch eine Anfrage eingeleitet (INVITE, BYE, REGISTER, …). Darauf können provisorische Statusantworten (z.B. 100 TRYING, 180 RINGING) folgen. Eine finale Statusantwort (z.B. 200 OK, 404 NOT FOUND) beendet die Transaktion. Manche Transaktionen (hier von Interesse: INVITE) bauen bei positiver Beantwortung der Anfrage einen Dialog auf, in dessen Kontext weitere Transaktionen stattfinden können. Bestimmte Transaktionen (hier: BYE) bauen einen Dialog wieder ab. Zum Aufbau von Transaktions- und Dialogstatus werden in den entsprechenden Systemen Zustandsautomaten nach RFC 3261 verwendet.

Der Dialog stellt eine Kommunikationsbeziehung dar, in dessen Kontext mit Hilfe des SDP (Session Description Protocol) die Parameter der Mediensession ausgetauscht werden. Die Parameter sind Art und Anzahl der Medien, mögliche Audio- und Videocodierungen, sowie Endpunktadressen der RTP-Ströme. SDP Nachrichten werden innerhalb von SIP Nachrichten übertragen. [4]

Die SIP Spezifikation definiert verschiedene Rollen, die ein SIP Netzelement einnehmen kann. Dabei wird darauf hingewiesen, dass es sich um logische Rollen handelt und nicht um konkrete Implementierungen. Ein an der SIP Kommunikation beteiligtes System kann je nach Konfiguration (auch pro Verbindung) eine andere Rolle einnehmen. In der Praxis werden solche Netzelemente oft mit weiteren Funktionalitäten kombiniert.

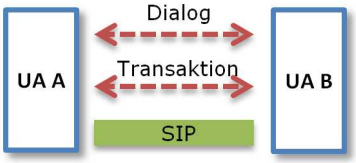

Bild 3 SIP User Agent

4.1 User Agent

Die Rolle User Agent (UA) stellt einen Signalisierungsendpunkt dar. Diese Rolle ist beispielsweise in SIP-Telefonen implementiert. Ein UA ist in der Lage Anfragen und Statusantworten zu erzeugen sowie Anfragen und Antworten zu verarbeiten. Stellt er eine Anfrage (z.B. INVITE), nimmt er die Rolle User Agent Client (UAC) ein. Eine Anfrage erhaltend nimmt er die Rolle User Agent Server (UAS) ein, erzeugt gegebenenfalls eine entsprechende Antwort und schickt diese an den UAC. Der UA terminiert Transaktionen und Dialoge (**Bild 3**).

4.2 Endgerät

Das Endgerät, also ein SIP Telefon, ist nicht explizit Teil der Spezifikation. Es besteht aus einer Signalisierungs- und einer Medienkomponente, die beide von einer übergeordneten Logik inklusive Benutzerschnittstelle gesteuert werden. Die Signalisierungskomponente kann mindestens die Rolle User Agent einnehmen. Diese konfiguriert auch die Mediensession, indem SDP Nachrichten verarbeitet und erzeugt werden. Die Medienkomponente übersetzt Audio und Videodaten in entsprechende RTP Datenströme und umgekehrt. Am Endgerät sind sowohl Signalisierungs- und Mediendaten vorhanden, so dass diese hier extrahiert und anderen Anwendungen zur Verfügung gestellt werden könnten.

4.3 Back-to-Back User Agent

Der Back-to-Back User Agent (B2BUA) besteht aus einer Konkatenation zweier User Agents und dient somit als Vermittler zwischen zwei Kommunikationspartnern.

Prinzip bedingt stellt er zwei Signalisierungsendpunkte dar und es werden zwei Kommunikationsbeziehungen (Dialoge) aufgebaut (**Bild 4**). Genau wie die Rolle User Agent hat der B2BUA die Möglichkeit die Mediensession zu konfigurieren. Jedoch terminiert das System, das die Rolle B2BUA übernimmt, in vielen Fällen nicht die Mediendaten. Eine Extraktion von Signalisierungsdaten inklusive zugehöriger Mediendaten ist nicht immer möglich.

Telekommunikationsanbieter setzen am Übergang des eigenen Netzes zu Fremdnetzen häufig sogenannte Session Border Controller (SBC) ein. Diese bestehen aus einem B2BUA und terminieren zusätzlich die Medienströme. Der Provider hat die volle Kontrolle über ein- und ausgehende Verbindungen. Damit ist unter anderem die Umsetzung von Sicherheitszielen, verlässliche Gebührenabrechnung und Einflussnahme bei der Wahl der Mediencodecs möglich.

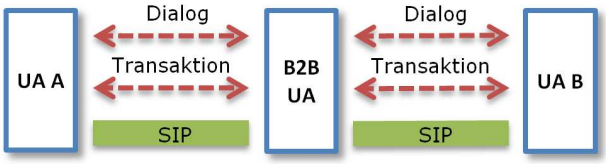

Bild 4 SIP Back-to-Back User Agent

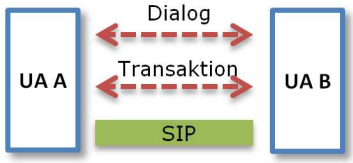

Bild 5 Stateful Proxy

Ein SBC besitzt in vielen Fällen eine sogenannte Lawful Interception (LI) Schnittstelle zur Extraktion von Signalisierungs- und Mediendaten zur Unterstützung von staatlichen Abhörmaßnahmen. Telekommunikations-anbieter sind verpflichtet entsprechenden gesetzlichen Anforderungen nach zu kommen. [5]

4.4 Proxy

Ein SIP Proxy dient dem Routing von SIP Nachrichten. Dazu werden DNS Server (Routing Richtung Zieldomaine), Location Server (Auffinden des Ziel- User Agents, der sich vorher registriert hat) sowie mögliche statische Konfiguration mit einbezogen. Es wird zwischen den Varianten stateless (zustandslos) und stateful (zustandsbehaftet) unterschieden. Der Stateless Proxy leitet SIP Nachrichten lediglich weiter, ohne den Dialog- oder Transaktionsstatus mitzuführen. Es wird also keinerlei Beziehung zwischen den Nachrichten hergestellt. Der Stateful Proxy hingegen arbeitet transaktionsterminierend (**Bild 5**). Da er aber den Dialogstatus nicht mitführt, ist er nicht zwingend an allen Transaktionen innerhalb eines Dialogs beteiligt. Beispielsweise ist die Rolle Stateful Proxy ohne erweiterte Logik nicht sicher in der Lage die Zeitdauer eines Telefongesprächs zu ermitteln. Dialogaufbauende Transaktionen (INVITE) und dialogabbauenden Transaktionen (BYE) können von dieser Rolle nicht in Beziehung gesetzt werden.

Laut Spezifikation hat ein Proxy keine Möglichkeit SDP Nachrichten zu verarbeiten oder anzupassen. Dies hat zur Folge, dass Signalisierungs- und Mediendaten unterschiedliche Wege zum Ziel- User Agent nehmen.

Das sogenannte SIP Trapezoid, wie in **Bild 6** dargestellt, zeigt ein mögliches Szenario bestehend aus zwei User Agents und zwei Proxys. Die Proxys werden zum Routen der Nachrichten einer initialen INVITE Transaktion verwendet (vgl. Bild 2, Transaktion 1).

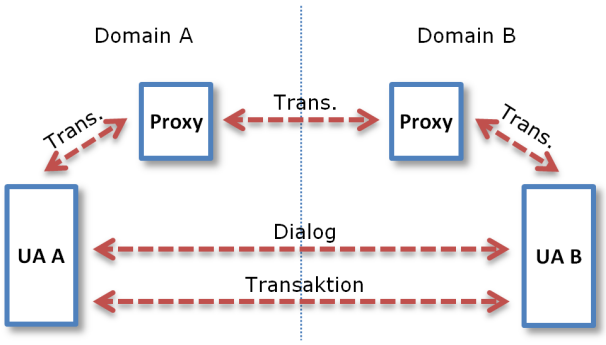

Bild 6 SIP Trapezoid, die Proxy-Server treffen ihre Routingentscheidung unter Einbeziehung von DNS und Location Servern

Aber der durch diese Transaktion aufgebaute Dialog und alle weiteren Transaktionen existieren zwischen den User Agents direkt. Die Proxys erhalten daher weder Kenntnis von einer nachträglichen Neukonfiguration der Medien–session, noch vom Verbindungsabbau. Die Mediendaten werden direkt zwischen den User Agents übertragen.

5 Datenextraktion

Je nach Anforderung kann die Extraktion an unterschiedlichen Systemen durchgeführt werden. Hier wird unterschieden, ob es sich um ein System handelt, das aktiv am Callflow beteiligt ist (Anwendungsschicht, verarbeitet SIP und eventuell RTP) oder ob es sich um ein transportorientiertes System (Switch / Router) handelt.

5.1 Anwendungsschicht-Extraktion

Systeme der Anwendungsschicht haben den Vorteil, dass diese ohnehin die benötigten Daten verarbeiten. Soll lediglich die Signalisierung extrahiert werden, kann dies an einem User Agent oder Back-to-Back User Agent durchgeführt werden. Diese Rolle ist im Gegensatz zum Proxy dialogterminierend, so dass alle durchgeführten Transaktionen erfasst werden. Zusätzlich könnten die entsprechenden Mediendaten extrahiert werden. In vielen Fällen verlaufen die Signalisierungsdaten nicht zusammen mit den Mediendaten über dieselben Systeme. Es sind jedoch verteilte Strukturen zur Extraktion möglich. Letztlich ist man auf vorhandene und entsprechend nutzbare Schnittstellen begrenzt. Die zur Extraktion verwendeten Systemressourcen können sich zudem negativ auf die Leistungsfähigkeit des VoIP Netzwerkes ausüben.

5.2 Extraktion am Monitoring Port

Bei der Extraktion am Monitoring Port eines Routers oder Switches wird eine exakte Kopie des Netzwerkstroms verarbeitet. Der Vorteil gegenüber der Anwendungsschicht- Extraktion ist die Unabhängigkeit von vorhandenen Systemen. Neben der einfachen, nachträglichen Integration in VoIP-Netze kann eine Überlastung oder gar Fehlfunktion des extrahierenden Systems keine negative Auswirkung auf das eigentliche VoIP-Netz haben.

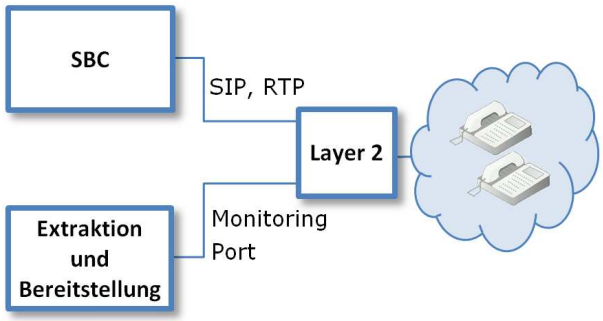

Bild 7 Extraktion am Monitoring Port (Switch oder Router)

Es müssen aber sowohl die Eigenschaften von SIP, als auch die Eigenschaften klassischer IP-Netze berücksichtig werden. SIP erzeugt oberhalb von IP ein Overlay-Netzwerk mit eigener Routingfunktionalität. Signalisierungs- und Mediendaten treten in größeren Netzen nur selten und an wenigen Punkten des Netzwerkes gemeinsam auf. Ein Beispiel bei dem das der Fall ist, ist der SBC als medienterminierender B2BUA. In diesem Fall kann der Switch oder Router, an dem dieses System angeschlossen ist, vollständige Signalisierungs- und Mediendaten an einem Monitoring Port zur Verfügung stellen.

Treten die Daten getrennt voneinander auf, so kann ein System die Kopie der Signalisierungsdaten auswerten und damit weitere Systeme steuern, die Zugriff auf die Mediendaten haben.

5.2.1 Interpretation der Signalisierung

Zur sinnvollen Extraktion der Mediendaten müssen vorhandene Kommunikationsbeziehungen vollständig bekannt sein. Es ist nicht ausreichend SIP Nachrichten über die Sender bzw. Empfänger IP Adresse zuzuordnen. In den meisten Fällen werden auf Netzwerkebene viele Verbindungen von ein und demselben System terminiert, beispielsweise wenn es sich um einen SIP Proxy oder einen B2BUA handelt. Andererseits können sich die IP Adressen sogar pro Transaktion ändern (vgl. **Bild 6**)

Das Extraktionssystem ist ein passives System, das lediglich die zwischen zwei SIP Elementen ausgetauschten Nachrichten interpretiert. Dazu baut es, genau wie die direkt an der Kommunikation beteiligten Systeme, den Dialog- und Transaktionsstatus mittels endlicher Automaten auf, die im SIP Standard spezifiziert wurden. Diese können jedoch stark vereinfacht werden, da die Timer zur Transportsicherung (sie steuern Paketwiederholungen und unter Umständen den Abbau einer Transaktion oder eines Dialogs) nicht verwendet werden können. Begründet ist das dadurch, dass dem passiven System nicht bekannt ist, welchen momentanen Wert die Timer der beteiligten Systeme haben. Außerdem sind die in der Spezifikation verwendeten Timer keine zwingenden Vorgaben, sondern können je nach Bedarf angepasst worden sein. Wird eine SIP Nachricht empfangen, muss das passive System davon ausgehen, dass die Nachricht den beabsichtigten Empfänger erreicht hat. Durch diese Vereinfachung fallen die beiden Zustandsautomaten der Client- und Server-Transaktion zu einem einzigen Automaten zusammen. Es muss lediglich zwischen einer INVITE Anfrage (**Bild 8**) und einer nicht-INVITE Anfrage (**Bild 9**) unterschieden werden.

Es empfiehlt sich jedoch einen eigenen Timer zu nutzen, mit dessen Hilfe im Falle von Paketverlust, beispielsweise durch Überlast des extrahierenden Systems, Transaktionen und Dialoge beendet werden können.

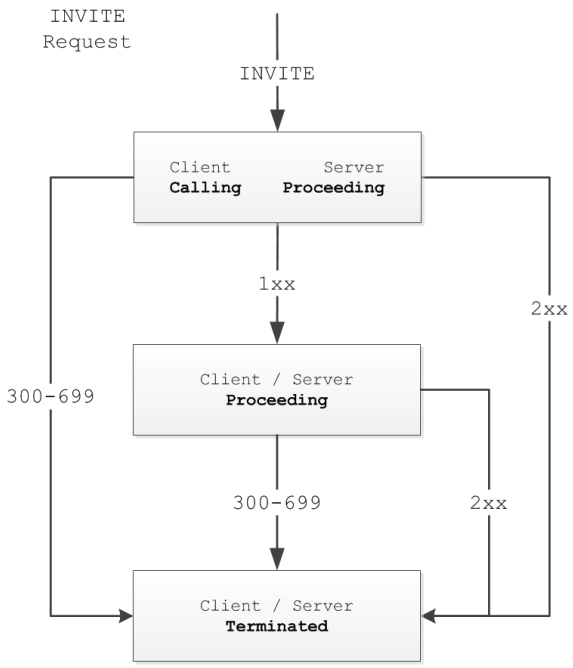

Bild 8 Transaktions-Automat des passiven Extraktionssystems für eine INVITE Anfrage. Zustandsänderungen sind abhängig von Statusantworten.

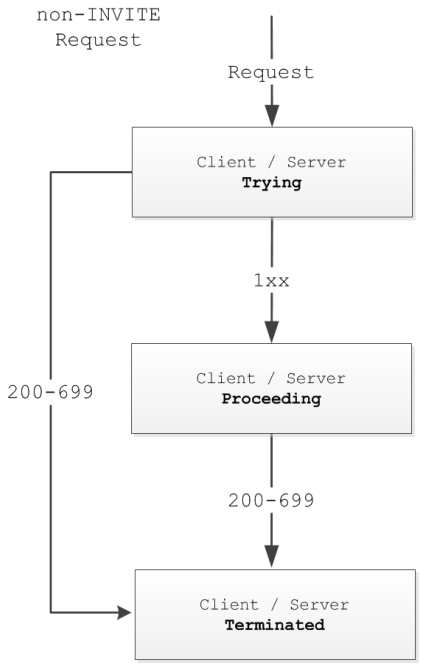

Bild 9 Transaktions-Automat des passiven Extraktionssystems für eine nicht-INVITE Anfrage.

5.2.2 Zuordnung Mediendaten

Beim Verbindungsaufbau werden innerhalb von SIP Nachrichten auch SDP Nachrichten übertragen. Für das Extraktionssystem sind die wichtigsten Informationen darin die Sockets, an denen die RTP-Ströme (Medienübertragung) terminiert werden. Da RTP ein unidirektionales Protokoll ist, werden bei einem normalen Telefonat mit bidirektionalem Audio zwei RTP-Ströme eingesetzt. Die Auswertung der SDP Nachrichten ist notwendig, da die SIP Endpunktadressen häufig wegen weiterer im Signalisierungspfad enthaltener Systeme (Proxy, B2BUA) nicht mit den RTP Endpunktadressen übereinstimmen.

Weiterhin werden über das SDP die möglichen Audiocodecs ausgehandelt (Offer/Answer Model [6]). Aber welcher Codec letztlich zur Codierung der Audiodaten verwendet wurde, ist im Header-Feld "Payload Type" eines jeden RTP Pakets hinterlegt. Die Nutzdaten des Pakets können dann entsprechend dekodiert als Audiodaten weiterverarbeitet werden.

6 Fazit

Verschiedene Anwendungen, beispielsweise Verfahren zur Identifikation und Abwehr von Telefon-SPAM, basieren auf der Auswertung von Signalisierungs- und Mediendaten aus VoIP Netzwerken. Die Verarbeitung einer Kopie der Netzwerkdaten bietet weitgehende Unabhängigkeit von produktiven Systemen, erfordert aber umfassende Kenntnisse über das Signalisierungsprotokoll und die Medienübertragung.

Im Projekt VIAT wurde eine Extraktions-Anwendung entwickelt, die mittels Monitoring Port eine Kopie des Netzwerkstroms analysiert und Signalisierungsinformationen sowie Audiodaten zur weiteren Verarbeitung bereitstellt.

Das Projekt VIAT wird vom Bundesministerium für Bildung und Forschung (BMBF) gefördert (Förderkennzeichen 1736X09).

7 Literatur

[1] Rosenberg J., Schulzrinne H., Camarillo G.: SIP: Session Initiation Protocol. Internet Engineering Task Force, Juni 2002. URL: http://www.ietf.org/rfc/rfc3261.txt

[2] Verfahren zur Identifikation und Abwehr von Telefon-SPAM, URL: http://viat.fh-koeln.de/

[3] Rosenberg J., Jennings, C.: The Session Initiation Protocol (SIP) and Spam. Internet Engineering Task Force, Category Informational, Januar 2008. URL: http://www.rfc-editor.org/rfc/rfc5039.txt

[4] Jacobson V., Perkins C.: SDP: Session Description Protocol. Internet Engineering Task Force, Juli 2006. URL: http://www.ietf.org/rfc/rfc4566.txt

[5] Trick, U., Weber, F.: SIP, TCP/IP und Telekommunikationsnetze, 4. Auflage 2009, ISBN 978-3-486-59000-5

[6] Rosenberg J., Schulzrinne H.: An Offer/Answer Model with the Session Description Protocol (SDP). Internet Engineering Task Force, Juni 2002. URL: http://www.ietf.org/rfc/rfc3264.txt

Autor

Dipl.-Ing. Bernhard Mainka studiert im Masterstudiengang "Technische Informatik" und ist wissenschaftlicher Mitarbeiter am Institut für Nachrichtentechnik der Fachhochschule Köln.

Kontakt: bernhard.mainka@fh-koeln.de