ulm university universität
uulm

Fakultät für
Ingenieurwissenschaften
und Informatik
Institut für Organisation und
Management von Informations-
systemen

Dezember 2015

Steuerung eines Fussballroboters mit mobilen Endgeräten

Bachelorarbeit an der Universität Ulm

OMI-2015-12-31

Vorgelegt von:
Patrick Lutz

Gutachter:
Prof. Dr. Stefan Wesner

Betreuer:
Dr. Lutz Schubert

Fassung 26. Oktober 2015

Satz: PDF-LaTeX 2ε

Abstract

Automation und eingebettete Systeme gehören in der heutigen Zeit bereits zum Standard. In der Industrie sind Roboter schon lange nicht mehr wegzudenken. Sie erledigen sowohl hoch präzise, als auch robuste, kraft-aufwendige Arbeiten wesentlich genauer und schneller als ein Mensch es je können wird. Ein weiteres Einsatzgebiet von Robotern sind für Menschen unzugängliche oder gefährliche Orte, wie zum Beispiel mit Gas gefüllte Höhlen oder sogar Orte außerhalb der Erdatmosphäre (zB. Mars-Rover). Doch ein weiterer, immer stärker anwachsender Anwendungszweig von Robotern ist die Unterhaltungsbranche. Diese Bachelorarbeit beschäftigt sich mit der Ansteuerung und Bedienbarkeit eines eigens entwickelten Fussballroboters.
Die Konstruktion der akkubetriebenen Roboter inklusive Softwareschnittstelle ist Thema einer weiteren Bachelorarbeit, auf die hier nur verwiesen wird.

Ziel dieser Bachelorarbeit ist es, ein Fussballspiel zwischen zwei Benutzern zu ermöglichen. Hierbei bilden Smartphones oder andere Computer mit einem Webbrowser die Fernsteuerung für die Roboter. Über diese Fernsteuerung ist es möglich die Roboter zu steuern und mithilfe eines am Roboter befestigten Schussapparats, ein Tor zu erzielen. Am Spielfeldrand sind wie im Fussball üblich Tore aufgestellt. Fällt ein Tor, findet eine Torerkennung statt und der entsprechende Spielstand wird an dem Bediengerät angezeigt. Außerdem findet eine Kameraübertragung von den Robotern zu den Bediengeräten statt, sodass das Spiel auch gespielt werden kann, ohne sich in unmittelbarer Nähe der Roboter zu befinden. Ebenfalls Teil der Arbeit ist es zu gewährleisten, dass das Spiel auch ohne manuelles Eingreifen spielbar bleibt. Dafür ist es notwendig, dass die Roboter bei niedrigem Akkustand selbstständig die Ladestation aufsuchen und sich automatisch laden. Damit die Roboter frei und ohne Hindernisse fahren können, findet die Kommunikation über Funk statt. Hierbei wird eine Client-Server Architektur gewählt, die es erlaubt, unabhängig von den Robotern unterschiedliche Bediengeräte zu implementieren. Ein weiterer Vorteil dabei ist, dass auch eine Kommunikation zwischen Server und Roboter stattfinden kann, ohne dass eine Fernbedienung verbunden ist, was für das selbstständige Anfahren der Ladestation wichtig ist. Das Finden der Ladestation funktioniert mittels einer Infrarot-LED, die impulslängenmodulierte Signale sendet.

Inhaltsverzeichnis

1 Einleitung

Die Geschichte der Roboter reicht bis in die Antike zurück. Schon dort gab es erste Versuche mit Automaten, die zum Beispiel Musik spielen sollten oder automatisch Theater spielen konnten. Mit dem Verlust der antiken Kulturen gingen jedoch auch die Kenntnisse über die Automation verloren. Erst im 13. Jahrhundert wurde ein Buch eines arabischen Ingenieurs bis in die westliche Welt bekannt, was Gerüchten zufolge auch Leonardo da Vinci für die Automation inspiriert haben soll. Roboter wie wir sie kennen gibt es erst seit Mitte des 20. Jahrhunderts. Der technische Wendepunkt, der mit der Erfindung des Transistors kam, machte es erst möglich, elektrische Schaltungen in einer Größe zu fertigen, die für Roboter notwendig ist. Seit diesem Zeitpunkt liegt ein wertvoller und nicht mehr wegzudenkender Industriezweig auf dem Gebiet der Robotik. Mit dem Einstieg der Robotik in die Industrie wurde die Produktion um ein vielfaches schneller und präziser, als sie von Menschenhand je vorgenommen werden könnte. Auch in Bereichen oder Umgebungen, die für Menschen lebensgefährlich oder ohnehin lebensunmöglich sind, spielen Roboter eine bedeutende Rolle. Ein Beispiel hierfür ist der Mars-Rover. Der Mars-Rover ist ferngesteuertes Fahrzeug, welcher für die Marsforschung verwendet wird und größten Teils von der Erde aus gesteuert wird. Wie die meisten ferngesteuerten Fahrzeuge ist er mit einer Vielzahl von Sensoren und Werkzeugen ausgestattet.

Diese Bachelorarbeit beschäftigt sich mit dem Thema der Steuerung von mobilen Robotern mit dem Ziel, ein Roboter-Fussballspiel zu ermöglichen. Auf einer Tischplatte in der Universität sollen zwei Roboter platziert werden. An den beiden Enden der Tischplatte werden Tore montiert, ähnlich wie man es beim Tischfussball kennt. Mithilfe von mobilen Endgeräten sollen nun die Roboter angesteuert werden können und ein Zwei-Spieler Fussballspiel ermöglicht werden. Die Roboter verfügen über einen Schussapparat, der es möglich macht den Ball zu beschleunigen. Sobald der Ball in ein Tor befördert wurde, wird ein Tor automatisch erkannt und auf den mobilen Endgeräten, welche die Steuercontroller bilden, angezeigt. Diese Steuercontroller können entweder ein Android-Gerät oder jedes beliebige andere Gerät sein, das über eine Tastatur und einen Webbrowser verfügt. Über diese Steuercontroller ist es möglich die Roboter zu navigieren und einen Schuss zu tätigen. Hierfür werden verschiedene Steuerarten bereitgestellt. Die Roboter bewegen sich vollkommen kabellos, was die Verwendung eines Akkus notwendig macht. Sobald der Akku einen Mindestprozentsatz unterschreitet, wird automatisch die Ladestation angefahren, damit sich der Akku des Roboters selbstständig, also ohne menschliches Eingreifen, lädt. Die Fertigung und Implementierung der Roboter ist Teil einer anderen Bachelorarbeit, auf die hier nur Oberflächlich eingegangen wird.

2 Grundlagen

UDP

Beim User Datagram Protocol handelt es sich um ein verbindungsloses und in jeglicher Hinsicht ungeschütztes Protokoll, das auf der Transportebene (Schicht 4 im OSI-Schichtenmodell) arbeitet. Es bestehen keinerlei Mechanismen, die das korrekte Übertragen von Paketen gewährleisten. Hierzu zählen:

Sicherheit Die Sicherheit, dass Pakete überhaupt ankommen

Reihenfolge Die Reihenfolge, in der Pakete ankommen

FlowControl Ein Überlaufschutz des Empfängerspeichers (FlowControl)

CongestionControl Stauvermeidung in der Übertragungskette (CongestionControl)

Angriffe Schutz gegen Paketmanipulation von dritten

Ein UDP Header besteht aus 8 Byte. Mit diesen 8 Byte werden lediglich Source-Port, Destination-Port, Länge und die Checksumme übertragen. Dieser vergleichsweise kleine Header (vgl. TCP mit etwa 20 Byte), führt zu einem geringen Overhead während der Übertragung, auch bei kleinen Paketen. Nachdem ein Paket gesendet wurde erfolgt keine Bestätigung des Pakets vom Empfänger. Durch diesen Uni-Direktionalen Sendevorgang entsteht wenig Traffic im Netzwerk. Falls gewisse Sicherheitsmechanismen gewünscht sind, müssen diese in höheren Schichten implementiert werden.
Aufgrund dieser Eigenschaften wird UDP in Bereichen eingesetzt, in denen es auf hohe Über-

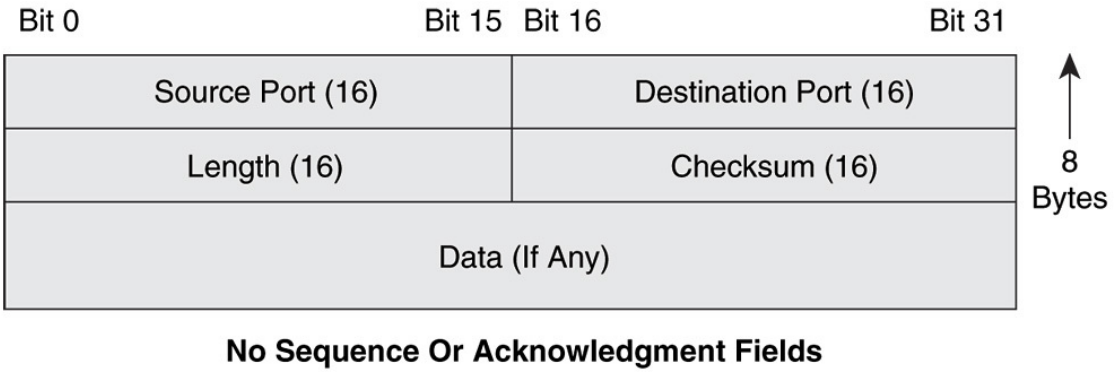

Abbildung 2.1: UDP Header

tragungsgeschwindigkeit ankommt und eventuelle Paketverluste zu verkraften sind, bzw. von höheren Schichten aufgelöst werden.

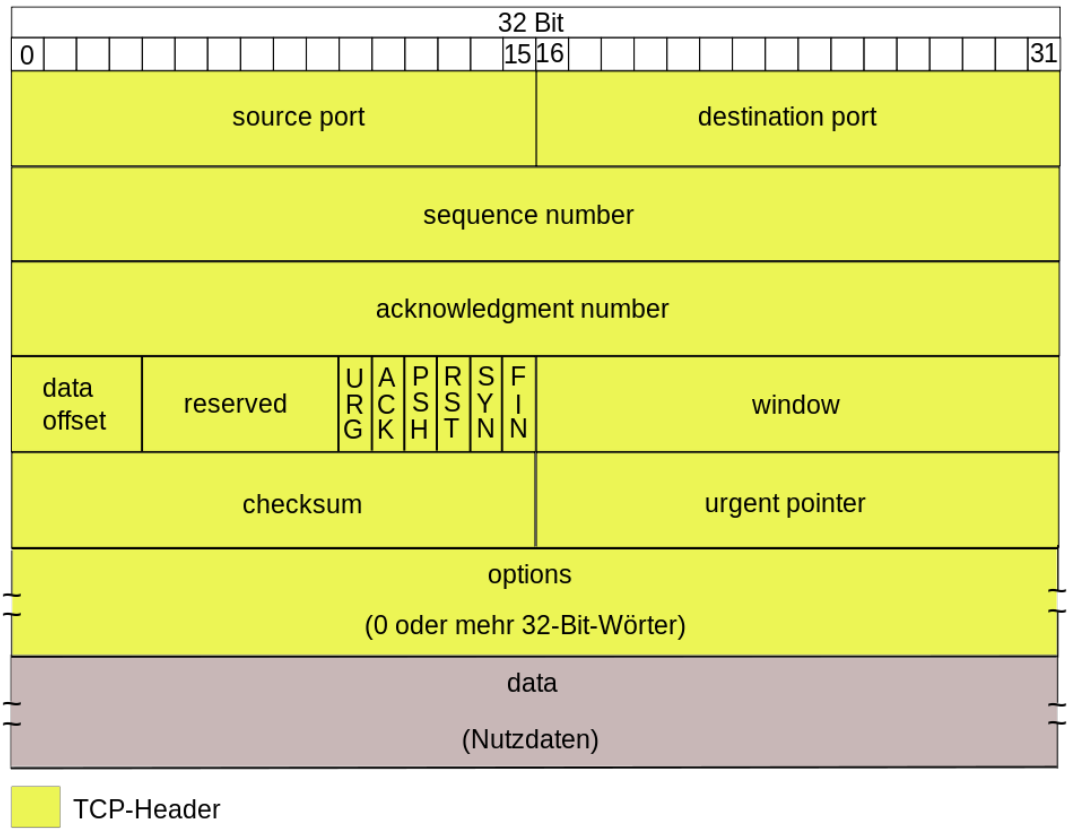

Abbildung 2.2: TCP Header

TCP

Beim Transmission Control Protocol handelt es sich um ein verbindungsorientiertes paketvermittelndes Protokoll. Mithilfe verschiedener Mechanismen wird sichergestellt, dass Pakete in der richtigen Reihenfolge ankommen, es zu keinen Staus kommt und dass Netzwerkknoten nicht überlaufen. Dadurch, dass das Protokoll verbindungsorientiert arbeitet, können beide Teilnehmer der Verbindung Daten ohne Anfragen senden.

Durch den größeren Header und dem größeren Traffic, der das Protokoll verursacht, wird TCP für Anwendungen verwendet, bei denen ein Paketverlust ausgeschlossen werden soll, dafür aber eine etwas höhere Latenz in Kauf genommen werden kann. Die Verbindung wird über einen 3-Wege-Handshake hergestellt. Hierbei sendet ein Teilnehmer eine Anfrage (syn), diese wird bestätigt (syn ack) woraufhin die Bestätigung erneut bestätigt wird. In diesem dritten Schritt werden meist bereits die ersten Nutzdaten mitgesendet.

Mithilfe von geordneten Sequenznummern und Acknowledgments werden alle empfangene Pakete bestätigt. Durch das Fehlen einer Sequenznummer erkennt der Empfänger den Verlust

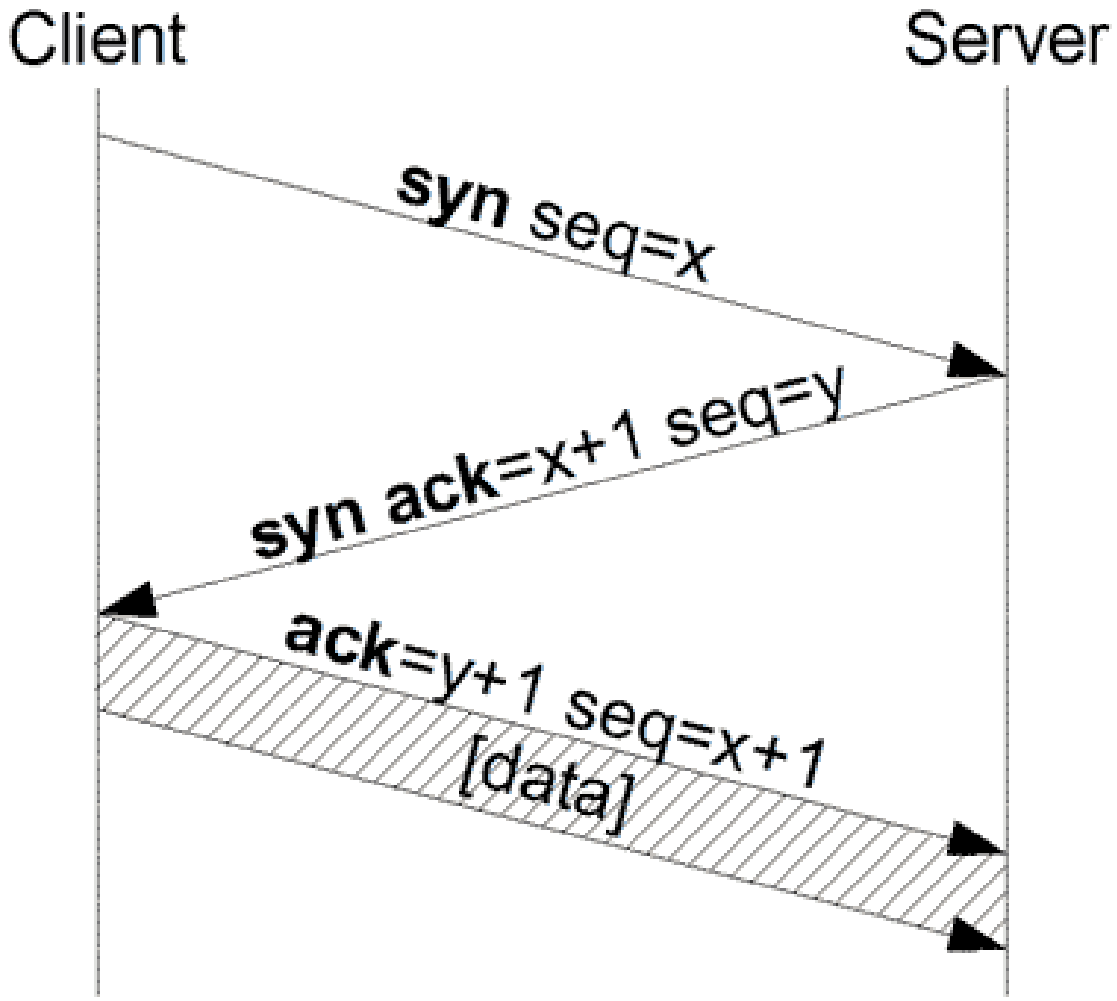

Abbildung 2.3: TCP 3-Wege-Handshake

eines Pakets und teilt daraufhin dem Sender mit, dass es nicht angekommen ist. Für den Fall eines Verlusts des Acknowledgments, hat der Sender einen Timer, welcher nach einiger Zeit eine Retransmission des Pakets veranlasst, sofern kein Acknowledgment ankommen sollte.

HTTP

Das Hypertext Transfer Protocol ist ein zustandsloses Datenübertragungsprotokoll, das auf der Anwendungsschicht arbeitet. Am meisten wird das Protokoll für den Aufbau von Internetseiten verwendet, also von einem Webbrowser, jedoch ist das Aufgabenfeld nicht darauf beschränkt.

HTTP arbeitet mittels zwei Befehlsarten, dem Request und dem Response. Möchte ein Browser eine Datei von einem Server laden, so sendet er einen Request mit dem Namen der Datei.
Die möglichen Befehle sind:

GET fordert eine Ressource auf dem Server an

POST sendet Daten zur weiteren Verarbeitung zum Server

HEAD fordert lediglich den HEADER eines Responses an, der auf einen GET-Request folgend würde

PUT lädt eine Ressource auf den Server

DELETE löscht eine Ressource auf dem Server (Wird kaum verwendet)

TRACE sendet die Anfrage, so wie sie empfangen wurde zurück

OPTIONS liefert eine Liste mit den vom Server unterstützten Optionen

CONNECT wird für SSL-Tunneling verwendet

Nachdem der Request beim Server eingegangen ist und verarbeitet wurde, sendet er einen Response. In diesem Response sind Informationen über Server und Datei enthalten, sowie die Nutzdaten, also die angefragte Datei. In Abbildung 2.4 ist eine solche Kommunikation dargestellt.
–

WebSockets

Das WebSocket-Protokoll ist ein Netzwerkprotokoll, das auf TCP und HTTP basiert. In diesem Protokoll ist es möglich, eine bidirektionale Verbindung zwischen einer Webanwendung und einem WebSocket-Server herzustellen. Im Gegensatz zu reinem HTTP ist es hierbei möglich, dass der Server ohne einen vorhergehenden Request des Clients, Daten an den Client sendet. Lediglich den Verbindungsaufbau muss der Client initiieren. Dies wird realisiert, indem die TCP-Verbindung nach dem Verbindungsaufbau nicht sofort geschlossen wird. Eine WebSocket URL wird über die beiden Schemata wss und ws definiert, was für verschlüsselte und unverschlüsselte Verbindungen steht.

Der Verbindungsaufbau funktioniert über einen Handshake, der wie in HTTP üblich über einen Request und anschließenden Response erreicht wird. Aufgrund der Tatsache, dass die

Abbildung 2.4: HTTP Kommunikation

HTTP-Header nur beim Verbindungsaufbau gesendet werden und dadurch der Traffic durch den HTTP-Header gering ist, wird das Protokoll hauptsächlich von Anwendungen verwendet, die regelmäßige Kommunikation zwischen Client und Server verlangen, wie zum Beispiel Online-Spiele.

Jetty Webserver

Jetty ist eine Java-Implementierung eines Webservers. Durch seine geringe Größe ist es leicht, ihn in andere Software zu integrieren. Außerdem unterstützt Jetty die Möglichkeit WebSockets aufzubauen.

3 Hauptteil

3.1 Software Architektur

Die Software-Architektur des Projektes definiert die Möglichkeiten, die man zur Implementierung der Anforderungen zur Verfügung hat. Deshalb war es wichtig, sich ausreichend Gedanken zu machen, um später im Projekt dann nicht feststellen zu müssen, dass die gewählte Architektur für die Anforderungen ungeeignet ist. Die Hauptanforderung an die Architektur ist selbstverständlich die Bereitstellung eines Kommunikationskanals zwischen Controllern und Robotern. Es muss möglich sein Richtungs- und Geschwindigkeitsänderungen seitens der Anwender an die Roboter mitzuteilen. Der Status des Roboters soll jederzeit ausgewertet und entsprechend darauf reagiert werden können. Hierzu gehören das automatische Anfahren der Ladestation bei geringem Akkustand und die Übertragung der Bilddaten. Außerdem sollen Zustandsänderungen des Spiels, die nicht vom Roboter oder Controller direkt ausgehen erkannt und korrekt verarbeitet werden.

Ausgehend von diesen Bedingungen, die die Architektur erfüllen muss, kamen folgende Architekturen in Frage:

Eine **Client-Server-Architektur** bei der ein Server zentral die Kommunikation verwaltet. Hierbei findet keine Kommunikation zwischen den Steuergeräten und den Robotern direkt statt. Die Torerkennung kommuniziert direkt mit dem Server, ohne dass Roboter und Controller zwangsweise etwas davon mitbekommen. Sowohl die Roboter, als auch die Controller nehmen die Rolle eines Clients ein. Dadurch ist es Möglich, dass eine Steuerung des Servers stattfinden kann, ohne dass ein Controller verbunden ist.

Erklärung

Ich, Patrick Lutz, Matrikelnummer 752774, erkläre, dass ich die Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den .....

Patrick Lutz