



Endbericht für das E-Learning Projekt *Visualisierung von Netzwerkalgorithmen mit HTML5*

Projekttitle: Visualisierung von Netzwerkalgorithmen mit HTML5

Projektcode: 2012.252

Projektleiter: Dr. Stefan Podlipnig

Beteiligte Personen: Johannes Kessler

Projektlaufzeit: 02/2012 – 06/2012

Titel der Lehrveranstaltung: Rechnernetze und Internettechnik

LV-Nummer: 703033

Beteiligte Fakultät: MIP

Beteiligtes Institut: Informatik



Projektziel

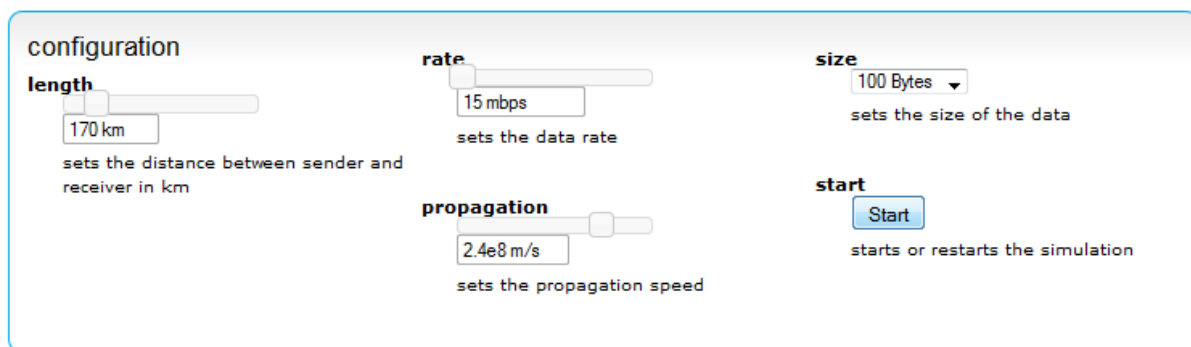
Die Lehrveranstaltung Rechnernetze und Internettechnik ist die erste Lehrveranstaltung im Informatik-Studium, die sich mit dem Internet und der Technologie dahinter beschäftigt. Da das Internet heute jeden Informatikbereich beeinflusst, ist ein genaues Verständnis der grundlegenden Algorithmen und Verfahren im Netzwerkbereich unabdingbar. Für dieses Verständnis muss aber der entsprechende Lehrstoff in einer dynamischen Form den Studierenden nähergebracht werden. Gerade im Internetbereich lässt sich die Dynamik vieler Verfahren nur unzureichend textuell oder mit statischen Grafiken darstellen und daher ist eine Visualisierung der dynamischen Abläufe im Lernprozess sehr hilfreich. Das Ziel dieser Arbeit war daher die Visualisierung von grundlegenden Verfahren im Netzwerkbereich (z.B. Übertragungsverzögerung, Go-Back-N-Protokoll, Selective-Repeat-Protokoll, Flusskontrolle) mit Hilfe von HTML5-Technologien.

Als Zielgruppe wurden zunächst Studierende der Lehrveranstaltung Rechnernetze und Internettechnik ausgesucht. Da diese Lehrveranstaltung im Sommersemester abgehalten wurde, konnten die erstellten Visualisierungen schon in diesem Semester verwendet und Rückmeldungen der Studierenden berücksichtigt werden.

Projektergebnisse und Technologien

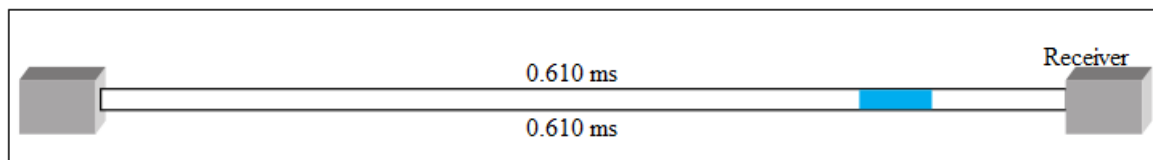
Transmission/Propagation Delay

Die erste Visualisierung illustriert den Unterschied zwischen Ausbreitungsverzögerung und der eigentlichen Übertragungsdauer von Daten über eine Verbindung mit bestimmten Eigenschaften.



The image shows a configuration interface for a simulation. It includes several controls:

- configuration**
 - length**: A slider set to 170 km, with the text "sets the distance between sender and receiver in km" below it.
 - rate**: A slider set to 15 mbps, with the text "sets the data rate" below it.
 - propagation**: A slider set to 2.4e8 m/s, with the text "sets the propagation speed" below it.
- size**: A dropdown menu set to 100 Bytes, with the text "sets the size of the data" below it.
- start**: A "Start" button, with the text "starts or restarts the simulation" below it.



coded by Johannes Kessler 2012 based on the [applet by David Granqier](#)

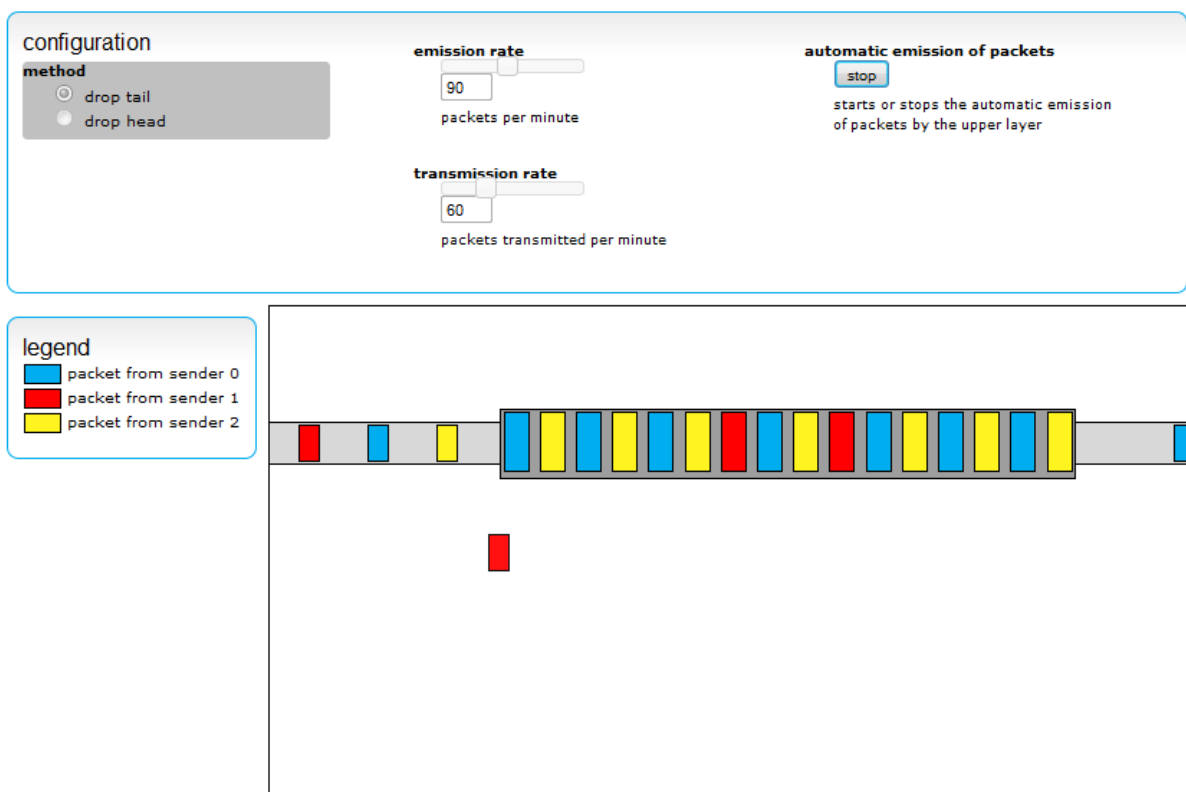
Für die verschiedenen Animationen wurden unterschiedliche Technologien verwendet. Zunächst wurde das sogenannte Canvas-Element getestet. Das Canvas-Element in HTML5 ist eine Zeichenfläche, auf der per JavaScript gezeichnet werden kann. Der Nachteil hierbei ist jedoch, dass die gesamte Ereignisbehandlung bei einer interaktiven Animation aufwändiger wird. Weiters muss der gesamte statische Teil der Animation entweder mittels JavaScript gezeichnet werden (Lesbarkeit des Codes wird verringert), oder als SVG oder Rastergrafik erstellt werden und dann im Canvas als Hintergrund ge-



zeichnet werden. Aus diesen Überlegungen heraus wurde daher die erste Visualisierung direkt als SVG gezeichnet und in die Webseite eingebettet.
Es wurde auch die JavaScript-Bibliothek Raphaël getestet, mit der man Vektorgrafiken deklarativ in JavaScript definieren kann und die dann das Rendering für verschiedene Browser übernimmt. Ein Grund dafür war, dass Raphaël auch ältere Versionen des Internet Explorers unterstützt, die keine native Unterstützung für SVG bieten. Beim Test ergab sich aber das Problem, dass bei vielen unabhängig voneinander bewegten Objekten, je nach Browser, eine nicht zufriedenstellende Framerate erreicht wurde.

Queuing and Loss

Die zweite Visualisierung illustriert das Verhalten einer Warteschlange z.B. in einem Router.



coded by Johannes Kessler 2012

Selective Repeat / Go Back N

Die dritte Visualisierung illustriert zwei wichtige Protokolle für die sichere Datenübertragung.

configuration

protocol

Go back N
 Selective Repeat
choosing a new protocol restarts the simulation

end to end delay

6000
time a packet takes from one station to the other

number of packets emitted per minute

70
the number of packets the upper layer tries to send per minute

window size

7
sets the window size for the windows

timeout

16000






automatic emission of packets

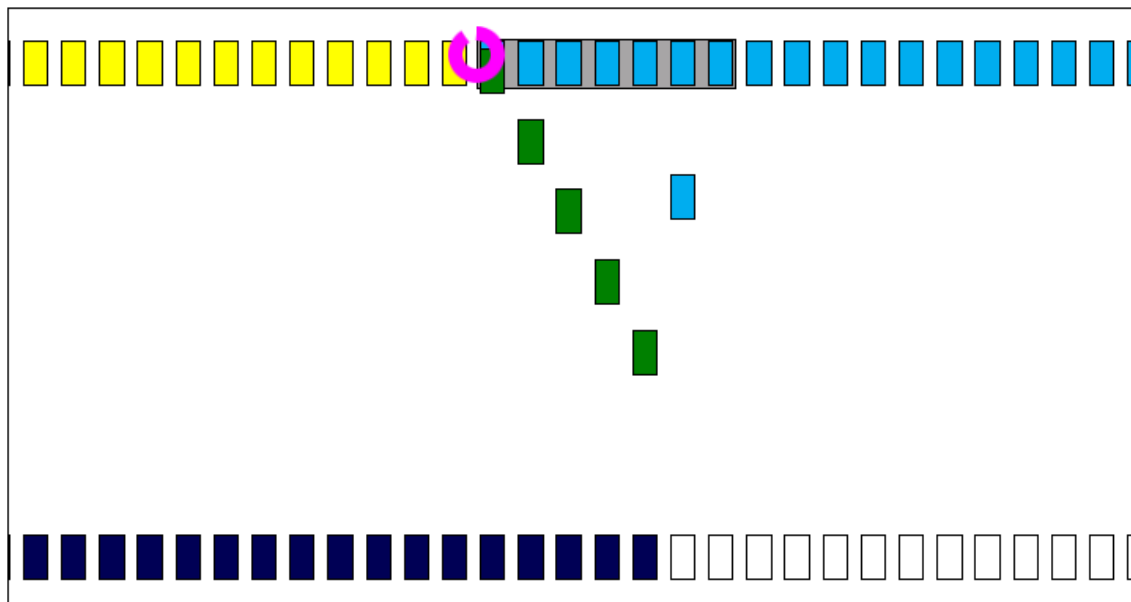
starts or stops the automatic emission of packets by the upper layer

scroll mode

Typewriter style
change the style the window scrolls

legend

-  no data received yet
-  data buffered (ready to send, delivered or sent but no ack received yet)
-  ack
-  transmission confirmed
-  data has been delivered to upper network layer



coded by Johannes Kessler 2012

Die zweite und dritte Visualisierung (*Queuing and Loss* und *Selective Repeat / Go Back N*) wurden ausschließlich mittels HTML-Markup umgesetzt, um von einer etwaigen Hardwarebeschleunigung des Browsers profitieren zu können. Für die Animationen der einzelnen Elemente wurde die Bibliothek jQuery verwendet, da dadurch die Verwaltung von Elementen und von Event-Handlern erleichtert wurde. Die Visualisierung der Timeouts wurde mittels Canvas realisiert.



Flow Control

Die vierte Visualisierung soll die Interaktion zwischen TCP-Sender und TCP-Empfänger illustrieren.

configuration

file size
8192 bytes
the size of the file the sender application wants to send to the receiver

buffer size
4096 bytes
size of the buffers

propagation time
2 sec
propagation time between sender and receiver

pause

pause or resume scrolling of the simulation

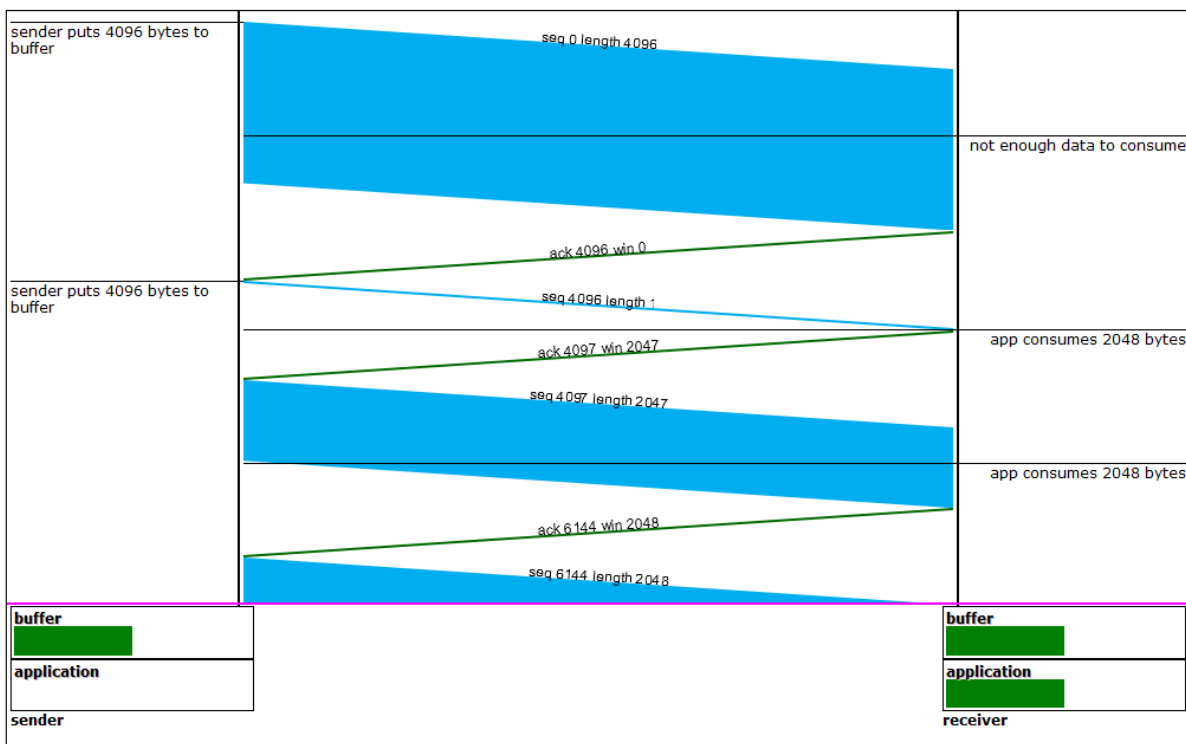
start

start or restart the simulation

legend

ack

data



coded by Johannes Kessler 2012

Bei dieser Visualisierung wurde der Darstellungscode möglichst generisch gestaltet, um eine Wiederverwendung in späteren Projekten zu erleichtern. Die einzelnen Balken und deren Beschriftungen wurden mittels Canvas dargestellt, da die Parallelogramme sonst mittels SVG oder HTML-Elementen umgesetzt werden müssten. Um dieses Aussehen zu erreichen, wurden dabei die Transform-Funktionen aus CSS verwendet, deren Unterstützung (noch) nicht bei allen Browsern vollständig gegeben ist.





Allgemeine Erfahrungen

Probleme

Die größte Schwierigkeit bei der Erstellung der Animationen war das Erreichen einer flüssigen Animation auf allen Browsern. Vor allem viele, mit jQuery unabhängig voneinander animierte Elemente, führten zu einem Einbruch der Framerate.

Dies wurde in einigen Visualisierungen dadurch gelöst, dass nicht jedes einzelne Element animiert wurde. Da sich alle Elemente mit der gleichen Geschwindigkeit bewegen, werden sie aktuell auf eine transparente Ebene gelegt. Nur diese Ebene wird dann bewegt.

Testläufe auf unterschiedlichen Technologien

Im Vorfeld wurde vereinbart, dass die Animationen auf sämtlichen modernen Browsern lauffähig sein sollten. Die Visualisierungen wurden daher auf folgenden Browsern getestet:

- Microsoft Internet Explorer 9
- Mozilla Firefox 10
- Opera Mobile 12
- Google Chrome 17
- Apple Safari 5.1
- Android Gingerbread Browser (bietet keine Unterstützung von Inline-SVG)

Um bei nicht unterstützten Browsern eine entsprechende Fehlermeldung anzuzeigen, wird aktuell für die Feature-Erkennung die JavaScript-Bibliothek Modernizr eingesetzt.

Änderungen zum ursprünglichen Plan

Die Änderungen zum ursprünglichen Plan liegen vor allem darin, dass nur ein Teil der Visualisierungen mittels HTML-Canvas erstellt wurde.

Fazit

Durch das im Sommersemester 2012 durchgeführte eLearning-Projekt *Visualisierung von Netzwerkalgorithmen mit HTML5* wurden Visualisierungen geschaffen, die in zukünftigen Lehrveranstaltungen verwendet werden können. Die erstellten Visualisierungen wurden schon in der Lehrveranstaltung Rechnernetze und Internettechnik von Studierenden getestet und für die Prüfungsvorbereitung verwendet. Außerdem wurden Rückmeldungen von den Studierenden in die aktuellen Versionen der Visualisierungen eingearbeitet.

Der Ablauf des Projekts gestaltete sich unkompliziert. Sowohl der beteiligte Mitarbeiter als auch die Abteilung für Neue Medien trugen zur reibungslosen Umsetzung dieses Projekts bei.

In Zukunft werden die erstellten Visualisierungen auch noch in den folgenden Lehrveranstaltungen verwendet werden:

- Programmierung von Sensornetzwerken (Bachelor, Wahlfach)
- Fortgeschrittene Kommunikationssysteme (Master, Wahlmodul)