

Wirtschaftlichkeit von energieeffizienten Netzkonzepten

Projektbericht

Veronika Lawrence, Carmen Scheer, Nicholas Cariss, Maximilian Junker,
Christian Keck, Stefan Ludowicy, Dominik Schneider

22. Juni 2016

Inhaltsverzeichnis

	Seite
1 Einleitung	4
2 Stand der Forschung	5
2.1 Carrier-Netzwerke	5
2.2 Möglichkeiten der Energieeinsparung in Netzwerken	5
2.3 Energie-effiziente Technologien	6
3 Problemstellung	6
3.1 Motivation	6
3.2 Ziele	8
4 Vorgehen	8
4.1 Modellierung	8
4.2 Softwareentwicklung	8
4.3 Entwicklung des Routing-Algorithmus	11
4.4 Komplexität bei der Entwicklung	11
5 Präsentation und Auswertung der Ergebnisse	11
5.1 Algorithmus	11
5.2 Die erstellte Software aus Ergebnissicht	14
5.2.1 Ergebnisse der Software-Entwicklung	15
5.2.2 Installation der Software	16
5.2.3 Benutzung der Software zum aktuellem Stand	18
5.2.4 Erweiterungsmöglichkeiten	18
5.3 Abschätzung des Energieverbrauchs	20
6 Diskussion und Erkenntnisse	20
7 Ausblick	20

Abbildungsverzeichnis

1	Die Entwicklung der Anzahl vernetzter Geräte	4
2	Weltweiter Stromverbrauch nach Sektoren, basierend auf [bibid]	7
3	Anteil von ICT am weltweiten Stromverbrauch, basierend auf [bibid] und [bibid]	7
4	ER-Diagramm	9
5	Programmablaufplan zur Anwendungslogik	12
6	Programmablaufplan zum übergreifenden Anteil des Routing-Algorithmus .	13
7	Rekursiver Anteil des Routing-Algorithmus	14
8	Schichtenarchitektur des EnergyNetSim (eigene Darstellung)	15
9	Schritte der Installation	16
10	Starten des MySQL-Server-Dienstes	17
11	Konfigurieren einer neuen Verbindung zum MySQL-Server	17
12	Das importierte Datenbankschema „energynetsimdb“	17
13	Hauptfenster nach dem Starten der Software	18
14	Dialoge „Select networks“ und „Settings“	18
15	Dialoge „Select networks“ und „Settings“	19
16	Ausgabe der Ergebnisse in Diagrammform	19

Tabellenverzeichnis

- 1 Parameter für das Einrichten des DB-Verbindung in der MySQL Workbench 17

1 Einleitung

Hier muss auch was stehen, oder? In der vorliegenden Arbeit... Sollten wir evtl die Unterkapitel-Aufteilung weg lassen?

Früher waren Geräte überwiegend fest verbunden, weshalb nur stationäres Handeln für Nutzer möglich war. Doch die Technik entwickelte sich hin zu mobilen Geräten. Nutzer konnten bald Geräte mit sich führen und unterwegs nutzen. Ein weiterer technischer Fortschritt war das Internet. Bereits 2003 waren circa 500 Millionen Geräte mit Internetzugang ausgestattet. Im Laufe der technischen Entwicklung wurde das vernetzte Handeln immer weiter vorangetrieben. Durch die Einführung des Internets der Dinge kam es dazu, dass Nutzer und Geräte vermehrt in ständigem Kontakt waren. Das Internet der Dinge bezeichnet „die Idee eines erweiterten Internets, welches neben klassischen Rechnern und mobilen Endgeräten auch beliebige physische Gegenstände [...] in seine Infrastruktur einbindet [...]“. Durch die Erwartung dieser Entwicklungsstufe, „Informationen aus Gegenständen abzurufen und aufzubereiten“, waren 2010 bereits 12,5 Milliarden Geräte mit Internetzugang versehen.

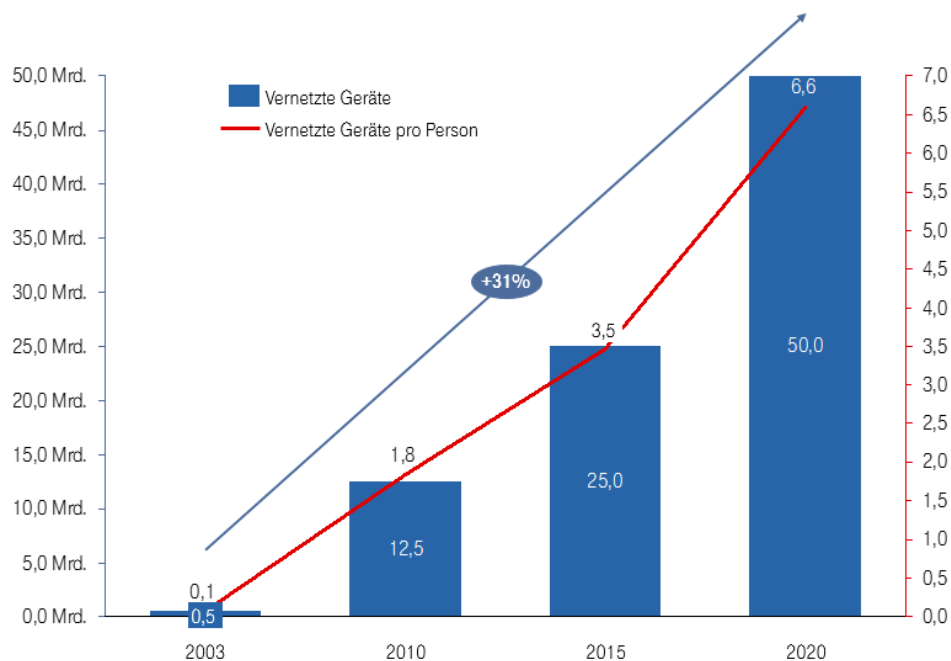


Abbildung 1: Die Entwicklung der Anzahl vernetzter Geräte

Zunehmend werden immer mehr physikalische Objekte mit Mikrokontrollern, Sendempfangern und geeigneten Protokollen für digitale Kommunikation ausgestattet. Dadurch steigt die Anzahl der mit dem Internet verbundenen Geräte, was Abbildung 1 illustriert. Prognostiziert ist, dass sich die Anzahl der mit dem Internet verbundenen Geräte bis zum Jahr 2020 auf 50 Milliarden erhöhen wird. Auf die Weltbevölkerung umgerechnet bedeutet das, dass jeder Mensch 6,6 vernetzte Geräte besitzen wird (siehe Abbildung 1). Mit diesem Durchbruch auf dem ITK-Markt entstehen neue Datenmengen, die früher noch nicht vorhanden waren. Diese Datenmengen müssen über Telekommunikationsnetze übertragen werden. Dadurch werden die Zugangsnetze und insbesondere die Kernnetze der Telekommunikationsanbieter stärker denn je belastet. Ein zweiter Trend der heutigen Informationsgesellschaft ist die rasant wachsende Nachfrage nach Bandbreite. Diese entsteht durch die Konvergenz von Medien und Telekommunikation. Orts- und zeitunabhängige TV-Nutzung, Video on Demand, Pay TV und HD sind beispielsweise treibende Fak-

toren für den Konsum von Bandbreite. Telekommunikationsnetze sind buchstäblich die Basis des täglichen Lebens und müssen mit immer mehr Datenverkehr umgehen können. In der logischen Konsequenz werden in Zukunft mehr Netzkomponenten verwendet, um den Anforderungen der vielen vernetzten Geräte und der Breitbandnachfrage gerecht zu werden. Das bedeutet auch einen Anstieg des Energieverbrauchs für die Betreibung der Telekommunikationsnetze. Bereits 2012 verbrauchten die herkömmlichen Rechenzentren in Deutschland jährlich über zehn Terawattstunden Strom, was etwa der Leistung von zwei Kernkraftwerken des Typs Brunsbüttel entspricht. Die wachsende Bevölkerung, der Klimawandel und die Verknappung der Energieressourcen stehen somit im starken Zielkonflikt. Eine Herausforderung ist es daher, die Telekommunikationsnetze umwelt- und ressourcenschonend zu gestalten.

2 Stand der Forschung

Definition Wirtschaftlichkeit, Carriernetz Betriebskosten, Wirtschaftlichkeit, etc. CAPEX OPEX Wirtschaftlichkeit Definieren Wir betrachten einen Teil der Betriebskosten, nämlich:

2.1 Carrier-Netzwerke

Gegenstand der Projektarbeit bildet ein hinsichtlich der Energieeffizienz und Wirtschaftlichkeit zu optimierendes Carrier-Netzwerk.

Unter einem Carrier „versteht man [...] eine Gesellschaft, die mindestens drei Übertragungswege betreibt, die über eine Vermittlungsstelle miteinander verbunden sein müssen“ (aus: **[carrier]**). Ein Carrier-Netzwerk stellt somit physikalische Transportwege und -verfahren zur Verfügung und bildet die Grundlage für sogenannte Value Added Services von Providern, welche auf den Carrier-Diensten aufsetzen **[fassnacht]**. „Bei den TK-Transportwegen unterscheidet man leitergebundene Verbindungen auf der Basis von Kupferkabeln oder Lichtwellenleitern sowie Funkverbindungen wie Satellitenverbindungen, Richtfunkstrecken und Rundfunkverbindungen“ (Aus: Ebd.).

Somit umfasst ein Carrier-Netzwerk nicht nur das physikalische Backbone-Netz, sondern auch das Zugangs- und Aggregationsnetzwerk. Im Rahmen des Projektes entfällt die Betrachtung der Optimierungspotentiale des Zugangsnetzes zugunsten einer ausführlicheren Simulation von wirtschaftlichen und energieeffizienten Netzkonzepten im Backbone- und Aggregationsnetzwerk. So bildet der Broadband Network Gateway (BNG) die niedrigste Netzelementebene. Multi-Service Access Nodes (MSAN), Digital Subscriber Line Access Multiplexer (DSLAM), enhanced NodeBs (eNodeB) und weitere Netzelemente der nächsttieferen Hierarchiestufe fallen somit aus der Betrachtung.

2.2 Möglichkeiten der Energieeinsparung in Netzwerken

Um den Energieverbrauch im Carrier-Netzwerk berechnen zu können, bedarf es an Werten der im Backbone verwendeten Netzkomponenten. Innerhalb der Simulation wird für die Berechnung des Gesamtverbrauchs auf die definierten Werte zurückgegriffen, die in einer Datenbank gespeichert sind. Ziel dieser Arbeit ist die Simulation und Berechnung des gesamtheitlichen Energieverbrauchs. Deshalb verwendet die Berechnung die von den beiden Wissenschaftler Ward Van Heddeghem und Filip Idzikowski in ihrer Veröffentlichung **[vanhedde]** zusammengetragenen Werte. Die Quelle beinhaltet zum einen das analytische Modell der Berechnung und zum anderen ein Datenblatt **[vanhsheet]** des Energieverbrauchs der unterschiedlichen Hersteller. Das Datenblatt gliedert die Geräte nach den

Unterschiedlichen Layer IP/MPLS, Ethernet, OTN, WDM - (OSI-Layer: 3-2-1-1). Zu beachten ist beim Verwenden der Werte, dass es sich um Werte unter typischen Lastbedingungen handelt, die sich nach der Kapazität der Komponente richtet und nicht nach dem aktuellen Durchsatz. Des Weiteren geben die Werte nur den Stromverbrauch für den Betrieb an, ein Verbrauch für Kühlung o.Ä. ist nicht enthalten. Der Gesamtverbrauch des Core-Networks ergibt sich aus der Summe aller Verbräuche der einzelnen Schichten.

$$P_{core} = P_{ip} + P_{ethernet} + P_{otn} + P_{wdm} \quad (1)$$

2.3 Energie-effiziente Technologien

Der aktuelle Stand der Forschung bietet verschiedene Konzepte und Technologien für das Einsparen von Energie in Carrier-Netzen. In dieser Arbeit sollen ausgewählte Konzepte und Technologien genutzt werden, um die Wirtschaftlichkeit von energieeffizienten Netzen zu analysieren. Betrachtet wurden im Rahmen der Literaturrecherche die folgenden Konzepte und Technologien:

Das erste Konzept sieht eine Vereinfachung des Netzes vor, welche durch eine geographische Aufteilung des Netzes in Submodule (Global - Kontinental - National - Regional - Zugang) erfolgen kann [aleksic2014].

Das zweite Konzept befasst sich mit der dynamischen Abschaltung unterlasteter Netzkomponenten. Tageszeitabhängige Schwankungen des Traffics in Netzen ermöglichen eine individuelle und dynamische Abschaltung von Switches und Links unter der Berücksichtigung der QoS-Bedingungen [aleksic2013]. Dabei werden Algorithmen zur Identifikation von individuell abschaltbaren Links verwendet [fisher].

Bei den Technologien beschränkt sich diese Arbeit auf optische leitungsvermittelnde Switches und das Hybrid Optical Switching (HOS). Optische leitungsvermittelnde Switches basieren auf mikro-elektromechanischen Systemen, die die geringste Menge an Energie benötigen und eine hohe Portanzahl besitzen. Das Hybrid Optical Switching verwendet eine Kombination aus optischen und elektronischen Netzknoten, die optische Leitungen, Bursts, und Pakete effizient durch Switches können. Durch die Kombination von langsamen und schnellen Switches, können Wellenlängenbereiche dynamisch zwischen zwei Switches geändert werden. Das temporäre Abschalten von ungenutzten Ports des schnellen Switches ermöglicht eine Energieeinsparung. [aleksic2013]

3 Problemstellung

Wie oben beschrieben existieren bereits einige Möglichkeiten, den Energieverbrauch in Carrier-Netzwerken zu senken. Allerdings werden diese Möglichkeiten noch nicht umfänglich in realen Netzwerken umgesetzt. Ein denkbarer Grund für diesen Rückstand realer Backbone-Netze sind wirtschaftliche Bedenken der Netzbetreiber. Im Folgenden wird die Motivation für Netzbetreiber beschrieben, den Energieverbrauch ihrer Netze stärker zu berücksichtigen, sowie die Ziele dieser Arbeit.

3.1 Motivation

Obwohl der Stromverbrauch der OECD-Staaten seit mehreren Jahren stagniert oder sogar leicht zurück geht, wachsen die globalen Verbräuche weiter an (vgl. Abbildung 2). Dies kann vor allem auf Schwellenländer zurückgeführt werden, die in kurzer Zeit ein rasantes wirtschaftliches Wachstum erleben.

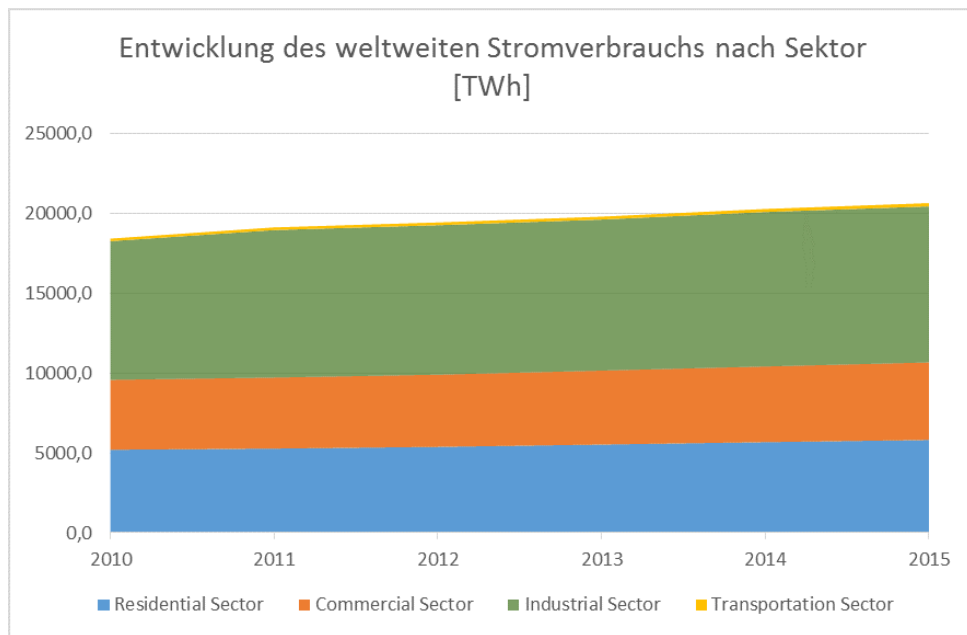


Abbildung 2: Weltweiter Stromverbrauch nach Sektoren, basierend auf [bibid]

Da aller Bemühungen zum Trotz die fossilen Energieträger Erdöl, Kohle und Erdgas weiterhin ca. 81% der weltweiten Energieerzeugung ausmachen [statista] und diese Ressourcen nicht regenerierbar sind, sind langfristig steigende Energiekosten ([iea2015], S. 40f) ein großes Risiko für ICT-Provider weltweit, die insgesamt mit steigenden Kosten zu kämpfen haben.

Steigende Energiekosten für sich genommen wären schon ein starkes Argument, Netze effizienter zu gestalten. Der Effekt der Kosten wird allerdings potenziert durch den Fakt, dass der Stromverbrauch von ICT von 2007-2012 stärker gewachsen ist als der globale Stromverbrauch (vgl. [vanhedde]). Schon 2012 betrug der Anteil von ICT am globalen Stromverbrauch etwa 5% (s. Abbildung 3)

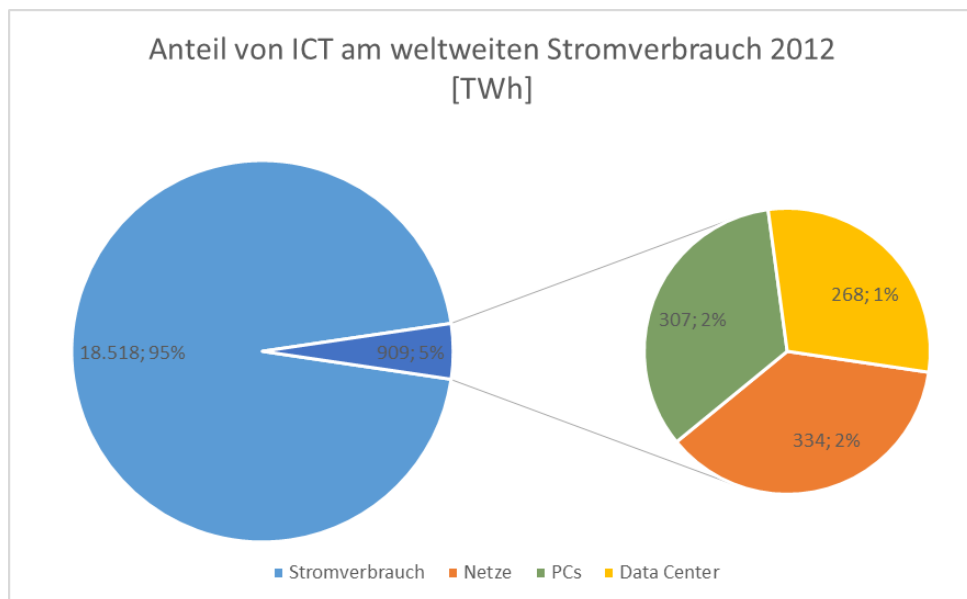


Abbildung 3: Anteil von ICT am weltweiten Stromverbrauch, basierend auf [bibid] und [bibid]

Seit dem ist der Anteil der Bevölkerung weltweit, der das Internet verwendet, von

20,6% (2007) auf 43,4% (2015) gestiegen [itu]. Dieses Wachstum wird sich in der nächsten Zukunft nicht verlangsamen. Weiterhin sorgen steigende Datenvolumen pro Nutzer für eine wachsende Netzlast. Welchen Effekt die Industrie 4.0 und das Internet of Things auf die Netze weltweit haben werden, lässt sich momentan noch nicht abschätzen. Eins jedoch steht fest: Das Netz der Zukunft wird mehr Daten zu bewältigen haben als jemals zuvor.

3.2 Ziele

Das Ziel der vorliegenden Arbeit ist es, abzuschätzen, wie viel Energie bzw. Kosten durch Verwendung energieeffizienter Technologien eingespart werden können.

Zur Erreichung dieses Ziels ist zum einen eine Literaturrecherche zu den bestehenden Technologien nötig, die es ermöglichen, den Energieverbrauch von Netzen zu senken. Die Ergebnisse dieser Literaturrecherche befinden sich in Kapitel 2.

Es soll eine Software entwickelt werden, die es ermöglicht, zwei hypothetische Netze miteinander zu vergleichen, zum einen ein konventionelles Netz, wie es heutzutage Stand der Technik ist, zum anderen ein energieeffizientes Netz, das die vorhandenen Technologien und Konzepte zur Effizienzsteigerung sinnvoll einsetzt. Anhand des abgeschätzten Energieverbrauchs der beiden Netze wird das Energiesparpotenzial sowie die möglichen Kosteneinsparungen durch den Betrieb des energieeffizienten Netzes ausgegeben. Die Ergebnisse der Softwareentwicklung sowie die Abschätzung des Energieverbrauchs befinden sich in Kapitel 5.

Die Softwarelösung wird als objektorientierte Java Desktop Anwendung mit einer Einteilung der Klassen in die drei Bereiche Model, View und Controller implementiert.

Der Prozess der Softwareentwicklung soll nach dem Wasserfallmodell ablaufen. Die Programmaufteilung und geforderte Funktionen sind bereits vor der Implementierung hinreichend bekannt, so dass die Anwendung eines agilen Entwicklungsmodells hier keine entscheidenden Vorteile bietet. Das genaue Vorgehen in der Softwareentwicklung wird im folgenden Kapitel erläutert.

4 Vorgehen

4.1 Modellierung

Zwei Netze modelliert (Quellen), um sie gegenüberstellen zu können. Techniken aus Kapitel 2 Energieeffiziente Technologien eingesetzt. Aufgrund der Recherche ist Simulation nötig.

4.2 Softwareentwicklung

Anforderungen an eine mögliche Simulationssoftware erarbeitet Entscheidung/Festlegung Entwicklungsmodell / Technologiestack / Systemlandschaft, weil.... Start Softwareentwicklung - Requirements - Spezifikation

- Design

Die zuvor beschriebene Modellierung zur Kalkulation und Simulation von energieeffizienten Netzen soll im nächsten Schritt softwaretechnisch umgesetzt werden. Als Vorgehensmodell hat sich das Projektteam an dem Wasserfallmodell mit den folgenden Entwicklungsstufen:

- Anforderungsanalyse
- Spezifikation

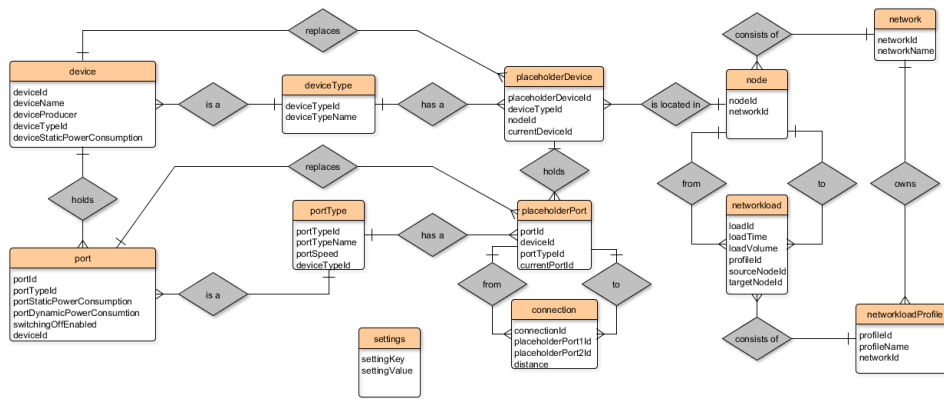


Abbildung 4: ER-Diagramm

- Systemdesign
- Programmierung / Implementierung
- Test

Die Analyse der Anforderungen an die Simulationssoftware erfordert eine zielgerichtete Recherche über bereits Vorhandene Technologien und Methoden sowie einer Definition von Funktionen, die das Simulationstool bereitstellen soll. Nach einer detaillierten Recherche konnten einige Anforderungen an das System gestellt werden. Das System soll ein statisches Netz mit einer festgelegten Anzahl an Verbindungen darstellen. Alle Daten zu Geräten, Netzen, Verbindungen und Netzprofilen sollen in einer MySQL-Datenbank gespeichert werden. Die Verwendeten Geräte sowie neue Netzprofile sollen über die Datenbank eingepflegt werden können. Für das Netzprofil sollen Traffic-Daten (Netzprofil, Last, Quelle, Senke, Uhrzeit) verwendet werden. Aus den Daten entsteht eine gleichbleibende Auslastungskurve, anhand der die Verbindungen für die jeweilige Uhrzeit berechnet werden kann. Um das Netz möglichst realistisch zu realisieren bekommen die einzelnen Verbindungen eine Länge zugeordnet. Anhand dieser Länge (mehr als 80 km) wird definiert, ob ein Verstärker hinzugefügt werden muss, um die Verbindung zu verbessern. Das definierte, statische Netz soll die Verwendung verschiedener Technologien und Methoden zur Senkung des Energieverbrauchs möglich machen. Zur Abschaltung Switches und Ports soll ein Algorithmus entwickelt werden, der auf der Basis des „Exhaustive Greedy Algorithm“ (QUELLE!!!) aufbaut. Im Zuge der Anforderungsanalyse und der Spezifikation der Software wurden die folgenden „functional“, „non-functional“ und „data“ Requirements erstellt: Functional Requirements:

- R1: Über die Datenbank kann ein Netz, bestehend aus Knoten (Geräte) und Kanten (Verbindungen) modelliert werden.
- R2: Ein Anlegen neuer Geräte und Gerätekonfigurationen ist über die Datenbank möglich.
- R3: Eine Veränderung von Geräteparametern ist über die Datenbank möglich.
- R3a: Ein Austausch von Gerätetypen ist über die Datenbank möglich
- (R3b: Ein Austausch von Gerätetypen pro Schicht ist über eine GUI möglich) — nicht in diesem Release

- R4: Das System berechnet anhand des modellierten Netzes, der eingestellten Gerätekonfigurationen der gewählten Energiespartechnologien und des vorliegenden Lastprofils den Stromverbrauch.
- R5: Das System ist in der Lage, aus dem Stromverbrauch anhand eines vorgegebenen Umrechnungsfaktors dessen Stromkosten zu berechnen.
- R6: Zur Vereinfachung von Berechnungen müssen Annahmen getroffen werden. Diese müssen in der DB hinterlegt werden.
- R7: Es werden generische Geräte basierend auf dem Dokument «Dokumentname einfügen» verwendet.
- R8: Als Algorithmus zur Abschaltung von Switches/Ports dient der «Exhaustive Greedy Algorithm»
- R9: Der simulierte Stromverbrauch im Laufe eines Tages pro Stunde soll graphisch ausgegeben werden.
- R10: Die Lastkurve wird graphisch dargestellt.

Non-functional Requirements:

- NFR1: Das System soll als Client-only Applikation vorliegen.
- NFR2: Das System verwendet eine MySQL-Datenbank als Persistenzschicht

Data Requirements:

- DR1: Für Verbindungen kann eine Länge vergeben werden.
- DR2: Für Gerätetypen können relevante Leistungs- und Verbrauchsparameter hinterlegt werden
- DR3: Ein Netz entsteht durch Verbinden von Geräteinstanzen und Zuordnung zu einer NetzID
- DR4: Annahmen werden durch einen eindeutigen Namen und einen Wert abgebildet
- DR5: Die Netzlast wird in Form von Datenpaketen mit den Attributen Volumen, Quelle, Senke, Uhrzeit, Profil-ID definiert

Während der Phase des Systemdesigns wurden einige Entscheidungen bezüglich der Softwareentwicklung getroffen. Als Programmiersprache für das Projekt wurde objekt-orientiertes Java aufgrund der bereits vorhandenen Entwicklungserfahrung im Team mit dieser Sprache und der sehr ausgeprägten Verfügbarkeit von kostenfreien Entwicklungstools und fertigen Bibliotheken ausgewählt.

Die Architektur besteht aus einer 2-Tier Kombination aus einer relationalen MySQL/MariaDB Datenbank und einer Fat-Client Anwendung. Die Datenbank wird dabei zur Speicherung der zur Erstellung eines Netzes notwendigen Daten, sowie Hardware-, Topologie- und Netzwerkkonfiguration verwendet. Die Anwendung hingegen wird zur Parameterkonfiguration, Berechnung der Zielwerte und Ausgabe der Resultate verwendet.

Die Vorteile dieser Architektur sind die gute Portabilität der resultierenden Lösung und die Ermöglichung einer einfachen und schnellen Entwicklung. Es müssen keine weiteren Anwendungsserverkomponenten je Entwicklungsumgebung installiert und konfiguriert werden. Die SQL-Datenbank verwendet eine übliche Datenbank-Standardsoftware und kann per SQL-Script vollautomatisch konfiguriert werden.

Implementierung

4.3 Entwicklung des Routing-Algorithmus

Zur Simulation des Netzwerkes, der Lastverteilung und des elektrischen Stromverbrauches des Netzwerkes ist ein iteratives Berechnen für einzelne Zeitabschnitte abhängig von der Netzlast, des Traffic-Ursprungs und des Traffic-Ziels notwendig.

Es stehen zur Berechnung des Stromverbrauchs je Zeitabschnitt generell mehrere Ansätze zur Verfügung. Der einfachste Ansatz wäre je Uhrzeit von einer bestimmten zuvor als Datenbankeintrag festgelegten Nutzungsauslastung je Verbindung auszugehen. Dies würde ein einfaches bestimmen der einzelnen Geräteauslastungen und anschließend addieren der Werte zu einem Gesamtstromverbrauch pro Zeiteinheit ermöglichen. Diese Ergebnisse könnten dann als Tabelle und Histogramm dargestellt werden.

Dieser Lösungsweg setzt allerdings eine sehr detaillierte und statische Modellierung eines Beispielnetzes und allen dazugehörigen Auslastungs- und Verbrauchswerten je Zeitabschnitt voraus. Aufgrund diesem hohen Konfigurationsaufwand wird ein Anwender folglich einen großen Zeitraum je Iteration auswählen, was zu einer sehr groben und realitätsfernen Simulation führen würde. Energiespartechniken wie das dynamische Abschalten von einzelnen Verbindungen und umrouten der Netzlast auf andere noch nicht optimal ausgelastete Verbindungen muss der Anwender in diesem Verfahren selbstständig je Iteration umsetzen.

Eine weitere Möglichkeit zur Berechnung der optimalen Stromverbrauchswerte bietet das automatisierte, in einem Algorithmus implementierte Berechnen der einzelnen Netzlast-Routing-Möglichkeiten und anschließend der Wahl des nach zuvor definiertem Regelsatz besten routings. Dieser Algorithmus bietet den Vorteil, dass der Anwender nur die Netzlast mit Iterationszeitraum, Quelle, Senke und Menge spezifizieren muss und die Software daraus selbstständig die beste oder eine gute Lösung errechnet. Dies erlaubt allerdings keine Anpassung des zu berechnenden N

4.4 Komplexität bei der Entwicklung

5 Präsentation und Auswertung der Ergebnisse

5.1 Algorithmus

Grundlegend ist das Ziel, für jeden Zeitabschnitt die beste erzielbare Lösung zur stromsparenden Bewältigung (Durchleitung) des Traffic von dem Eingangspunkt in das Netz (Quelle) bis zum Austrittspunkt (Senke) zu berechnen. Zur Vereinfachung wurde die Annahme getroffen, dass Über die Dauer eines Zeitabschnittes (einer Iteration) die durch den Algorithmus getroffene Routing-Entscheidungen und die Wahl der abzuschaltenden Hardware gleich bleiben. Diese werden je Iteration initial einmal neu berechnet. Durch die Verkürzung der Iterationsdauer und der dazu passenden Netzlast-Daten lässt sich bei geringem Aufwand ein genaueres, verfeinertes Modell durchrechnen, so dass ein realistischeres Resultat zu erwarten ist.

Je durch den Anwender in der Datenbank hinterlegtem statischen Netz wird eine Datenreihe bestehend aus einem Stromverbrauchswert je Iterationszeitraum berechnet. Um diese Einzelwerte zu berechnen, werden zuerst die Lasten aus jedem Netzlastprofil-Eintrag auf möglichst effizient auf die Netzwerkhardware und Verbindungen projiziert, und danach abschließend abhängig von den berechneten Geräteauslastungen die Teil-Stromverbrauchswerte addiert.

Mit dieser groben Vorgehensweise lässt sich das gesuchte Ergebnis je Modellnetzwerk inklusive dem zeitlichen Verlauf errechnen. Allerdings fehlt dazu noch die Antwort auf die

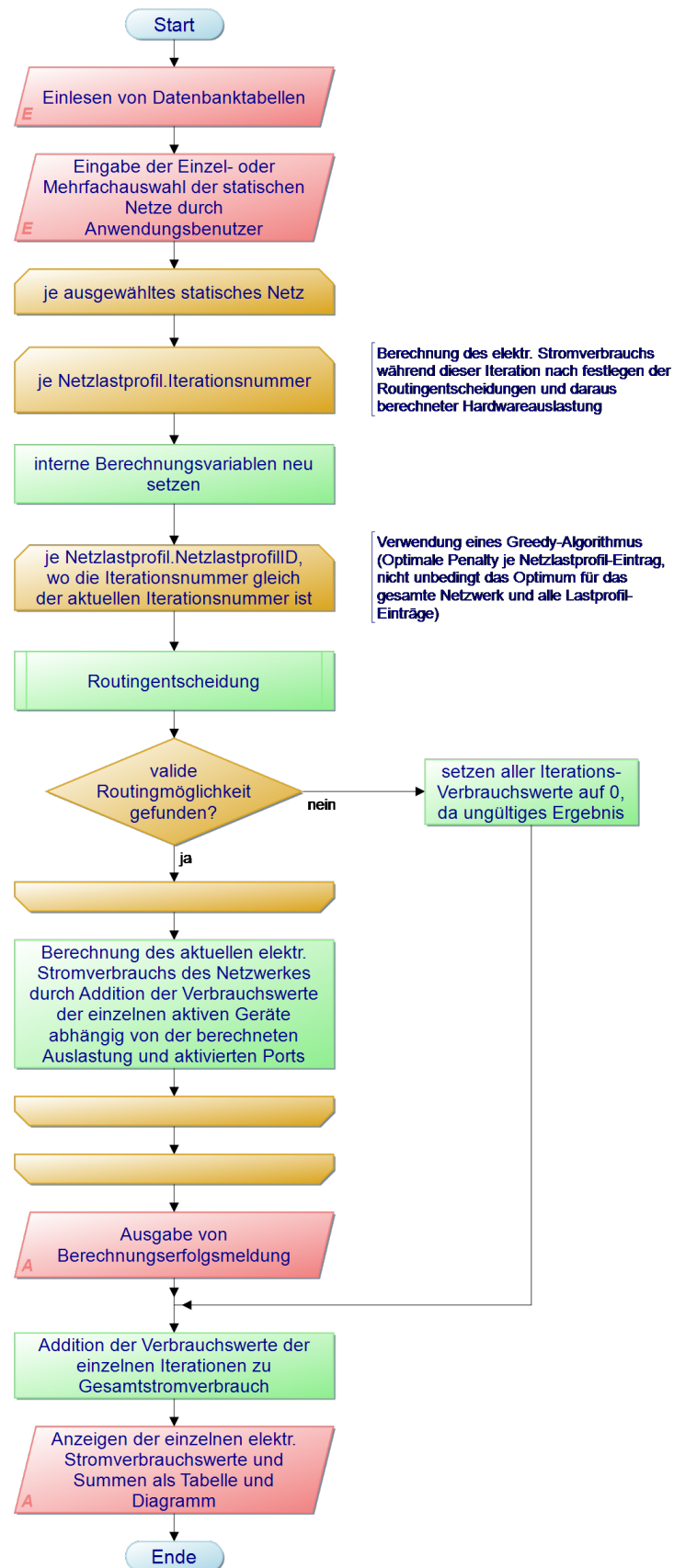


Abbildung 5: Programmablaufplan zur Anwendungslogik

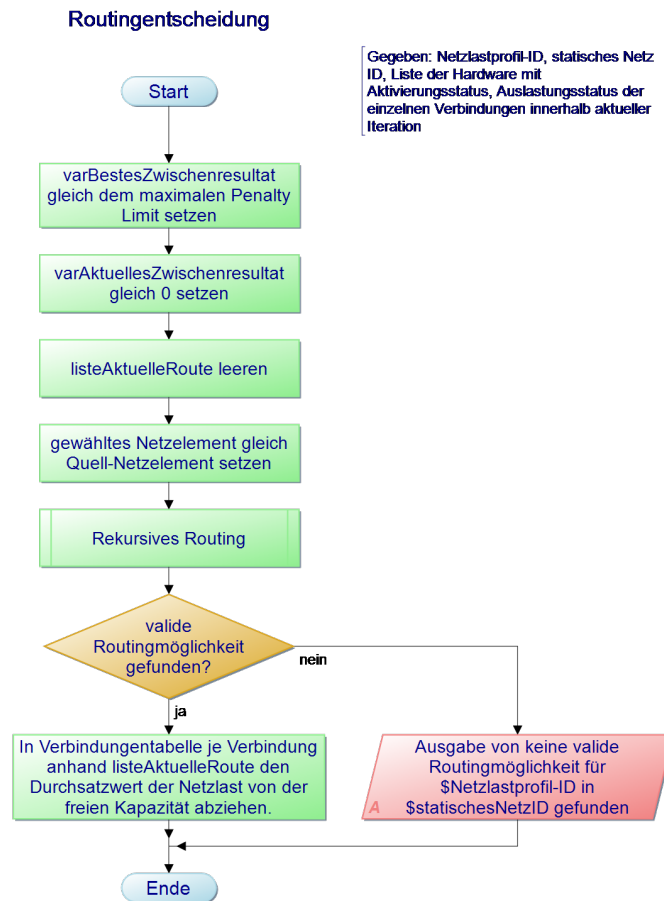


Abbildung 6: Programmablaufplan zum übergreifenden Anteil des Routing-Algorithmus

Frage, wie die effizienten Routing-Entscheidungen getroffen werden können. Der folgende Teil des Algorithmus beschreibt eine möglichen Lösung:

Um zu entscheiden, welches Routing für die einzelnen Netzlastprofil-Einträge je Iteration das beste Resultat liefert, wird je mögliche Route eine dynamische Penalty (Kosten-Faktor) berechnet. Anschließend wird für das Netzlastprofil die Route mit der niedrigsten errechneten Penalty, welche die Datendurchsatzgrenzen der Hardware und somit der einzelnen Verbindungen nicht überschreiten, als beste Route angenommen. Diese Kategorie von lokalen Lösungsverfahren nennt man Greedy Algorithmus. Dieser liefert nicht unbedingt das insgesamt beste Ergebnis, sondern berechnet nur das lokale Optimum für das betrachtete Teilproblem. Um eine schnelle praktische Ausführungszeit auch bei Berechnungen mehrerer Vergleichsnetze mit kurzen Iterationsintervallen zu erreichen beinhaltet dieser Algorithmus intelligente Entscheidungsfunktionen um schlechte Routen frühzeitig zu ignorieren und nicht zielführende Berechnungen wie sie beispielsweise bei Schleifenbildung auftreten zu verhindern.

Da die zur Verbindungsbewertung verwendete Penalty mehreren Faktoren wie Latenz, elektr. Stromverbrauch, Kapazität der Verbindung und auch An-/Aus-Status der Hardware berücksichtigen muss, und diese einzelnen Faktoren je nach Verwendungszweck des Netzes unterschiedliche Gewichtung haben, müssen die einzelnen Anteile mit vom Anwender der Simulationssoftware festgelegten Gewichtungsfaktoren multipliziert werden. Damit kann der Nutzer die Netzsimulation auf seine Anforderungen ansatzweise anpassen.

Alles in allem ist ein Regelsatz zur Penaltyberechnung je Iteration notwendig, an

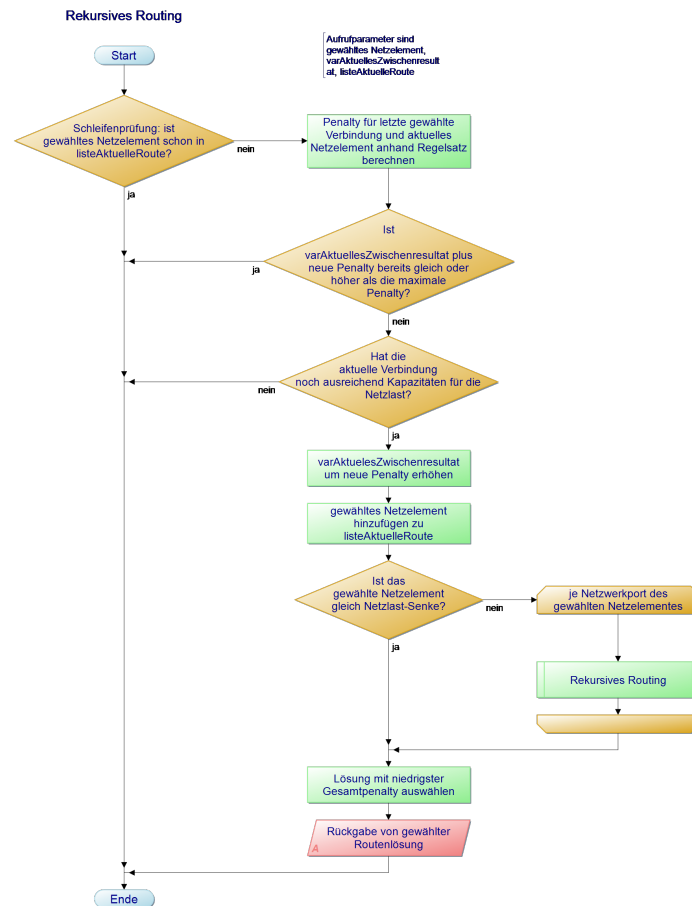


Abbildung 7: Rekursiver Anteil des Routing-Algorithmus

Hand welchem die Algorithmus-Implementierung jeweils versucht das lokale Optimum zum Routingproblem als Lösung zu finden. Der folgende Regelsatz stellte sich beim logischen durchspielen einer solchen Simulation als Grundlage heraus:

- Zum Anfang jeder neuen Iterations-Zeitphase sind alle Geräte und alle Ports soweit diese dies unterstützten ausgeschaltet
- Das Aktivieren eines Devices kostet Penalty (hoch)
- Das Aktivieren eines Ports kostet Penalty (niedrig)
- Das Ausschalten des Port-Energiesparmodus kostet keine Penalty, um vermeidbares aktivieren anderer Hardware / Ports zu vermeiden. Energiesparmodus-Capability zählt lediglich zur Berechnung des Stromverbrauchs.
- Um niedrige Paketlaufzeiten durch das Netz sicherzustellen, kostet jeder Hop / genutzte Verbindung eine weitere Penalty (hoch)

Einzelne Verbindungen haben zusätzlich zu dem dynamischen Anteil außerdem jeweils eine feste Penalty basierend auf Verstärkeranzahl + Länge -i statischer Wert in Datenbank.

5.2 Die erstellte Software aus Ergebnissicht

Im Rahmen der Software-Entwicklung wurde das Java-Programm „EnergyNetSim“ mit- samt Anbindung an eine MySQL-Datenbank realisiert. Die folgenden Unterkapitel be-

schreiben die erzielten Resultate und geben Hinweise auf Installation und Benutzung der Software für Endanwender.

5.2.1 Ergebnisse der Software-Entwicklung

Die erstellte Software „EnergyNetSim“ stellt ein Rahmenwerk zur Verfügung, in das die in den Kapiteln 4.3 und 5.1 beschriebenen Algorithmen zur Simulation von Netzauslastung, Datenverkehr zwischen zwei Netzknoten, dynamischem Routing und effizienzsteigernden Energiesparmechanismen eingefügt werden können. Im Programmcode und in der Dokumentation sind die dazu notwendigen Methoden bereits angelegt und gekennzeichnet. Das Kapitel 5.2.4 gibt dazu detailliertere Hinweise. Aufgaben wie die Selektion der zu betrachtenden Netze, die einfache Adaption von Parametern zur Simulation, die persistente Speicherung der Konfigurationen und Netzstrukturen und die graphische Ausgabe der errechneten Ergebnisse übernimmt der aktuelle Softwarestand bereits. Die Grundlage dafür bietet die Einbindung der frei verfügbaren Java-Bibliothek „jFreeChart“¹ und des offiziellen JDBC-Treibers² zur Verbindung mit der MySQL-Datenbank. Das Zu-

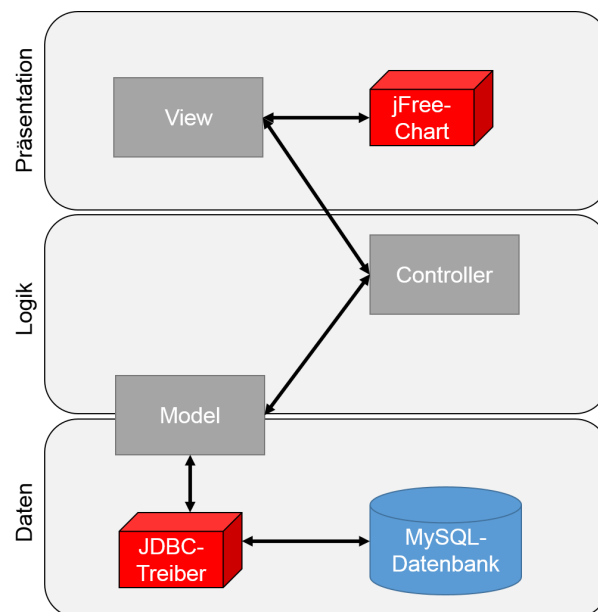


Abbildung 8: Schichtenarchitektur des EnergyNetSim (eigene Darstellung)

sammenspiel zwischen Java-Anwendung und der MySQL-Datenbank zeigt Abbildung 8. Bei der Entwicklung wurde ein „Model-View-Controller“ (kurz: MVC)-Entwurfsmuster gewählt, das die Trennung der Programmstruktur in die logischen Schichten „Datenhaltung“, „Programmlogik“ und „Präsentation“ ermöglicht. Das Programm wird mit den Datenbank-Skripten zur Erstellung der Datenbankstrukturen auf zwei Arten zur Verfügung gestellt:

- Als offenes Repository „EnergyNetSim/Simulator.git“ über den Versionsverwaltungsdienst GitHub: <https://github.com/EnergyNetSim/Simulator.git>

¹„JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart’s extensive feature set includes, a consistent and well-documented API, supporting a wide range of chart types; a flexible design that is easy to extend,[...] support for many output types, including Swing and JavaFX components, image files (including PNG and JPEG).[...] JFreeChart is open source or, more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications.“
[jdbc]

²Frei verfügbar unter <https://dev.mysql.com/downloads/connector/j/5.1.html>

- Als ausführbares Java-Archiv (JAR), das in der Anlage zum vorliegenden Bericht enthalten ist. Auf die notwendigen Schritte zur Installation der erstellten Software wird im nachfolgenden Kapitel eingegangen.

5.2.2 Installation der Software

Von den zwei vorgestellten möglichen Wegen, über die Simulationssoftware zu beziehen ist, wird im Anschluss die Inbetriebnahme mittels JAR-Datei beschrieben. Das Kapitel richtet sich somit vornehmlich an Endanwender. Die Einbindung des GitHub-Repositories ist vor allem für Entwickler interessant und erfordert daher Kenntnisse im Umgang mit der Versionsverwaltungssoftware Git, einer zusätzlichen Java-Entwicklungsumgebung, beispielsweise IntelliJ oder Eclipse. Des Weiteren muss das Java-Development-Kit in seiner aktuellsten Version installiert werden. Was gerade in Kurzform skizziert wurde, würde in einer ausführlichen Fassung den Rahmen des Berichts sprengen. Außerdem existiert eine ausreichende Zahl an Online-Communities und Anleitungen³ zur Verwendung der beschriebenen Programme, sodass aus einer erneuten Schilderung kein wissenschaftlicher Mehrwert entstünde. Bis das Java-Programm mittels JAR-Datei ausgeführt werden kann, sind die in Abbildung 9 abgebildeten Schritte erforderlich. Voraussetzung ist das Vorhandensein

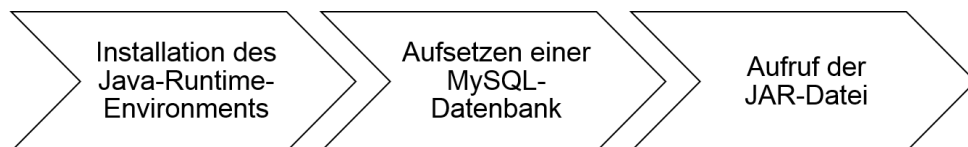


Abbildung 9: Schritte der Installation

der aktuellsten Version der Java-Laufzeitumgebung, kurz „JRE“. Innerhalb dieser Umgebung kann die Anwendung betriebssystemunabhängig in der Java-Virtual-Maschine (kurz: „JVM“) ausgeführt werden. Das Java Runtime Environment ist über die Webseite der Firma ORACLE⁴ zu beziehen und zu installieren. Weil die Java-Anwendung auf eine MySQL-Datenbank als Persistenzschicht zugreifen soll, wird zusätzlich ein lokaler MySQL-Server benötigt, der über die IP-Adresse 127.0.0.1 / „localhost“ und den Port 3306 erreichbar ist. Das Entwicklerteam empfiehlt den Einsatz des Open-Source-Datenbankservers „MySQL Community Server“, der ebenfalls von ORACLE⁵ unter der GNU General Public License kostenlos angeboten wird. Ebenfalls empfehlenswert ist die Verwendung des Datenbank-Modellierungs- und Manipulationswerkzeugs „MySQL Workbench“, das über denselben Weg bezogen werden kann. Nach der erfolgreichen Installation der beschriebenen Komponenten muss der SQL-Server gestartet werden, um die Datenbank anzulegen. Für Nutzer eines Windows-Betriebssystems sind dazu folgende Schritte erforderlich:

Starten des MySQL-Server-Dienstes: Durch Klick auf „starten“ wird der SQL-Server hochgefahren und der Status ändert sich in „Gestartet“.

Anlegen einer neuen Verbindung zum Server: Innerhalb der MySQL-Workbench wird eine Verbindung zum MySQL-Server mit den in Tabelle 1 aufgeführten Daten hergestellt.

³Auf der Webseite von GitHub findet sich eine Sammlung von Links zum Erlernen von Git: <https://help.github.com/articles/good-resources-for-learning-git-and-github/> Ebenso bietet Oracle eine ausführliche Java-Dokumentation: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

⁴Zum Zeitpunkt der Abgabe des Berichtes ist der Download über folgenden Link möglich: <http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

⁵Zum Zeitpunkt der Abgabe des Berichtes ist der Download über folgenden Link möglich: <http://dev.mysql.com/downloads/mysql/>

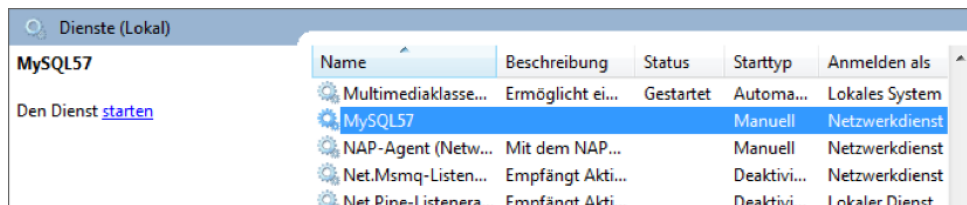


Abbildung 10: Starten des MySQL-Server-Dienstes

Parameter	Wert
Connection Method	Standard (TCP/IP)
Hostname	127.0.0.1
Port	3306
Username	root
Passwort	Wird bei der Installation angezeigt

Tabelle 1: Parameter für das Einrichten des DB-Verbindung in der MySQL Workbench

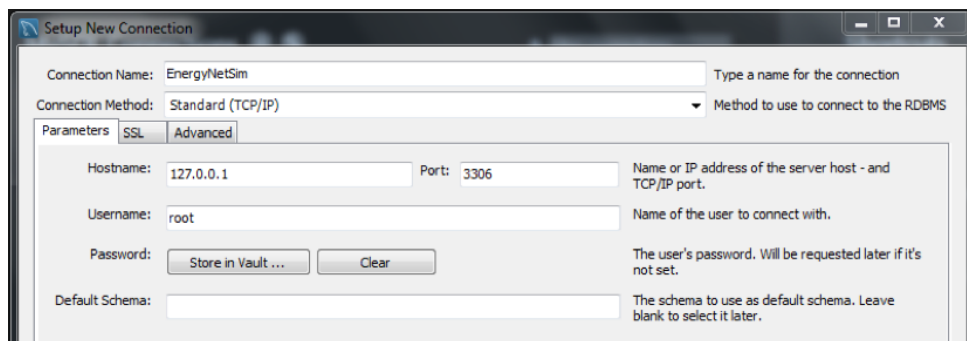


Abbildung 11: Konfigurieren einer neuen Verbindung zum MySQL-Server

Importieren des Datenbank-Schemas: Das Initialisierungsskript der Datenbank, welches sich im Anhang des Projektberichts angehängt ist, kann daraufhin über Server \rightarrow Import Data \rightarrow Import from Self-Contained File auf dem neuen MySQL-Server ausgeführt werden. Im Ergebnis existiert nun das Schema „energynetsimdb“.

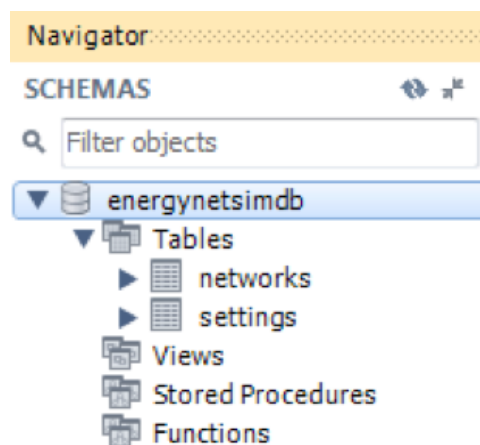


Abbildung 12: Das importierte Datenbankschema „energynetsimdb“

Ausführen der JAR-Datei: Durch Doppelklick auf die Datei „EnergyNetSim.jar“ wird das Programm gestartet. Im weiteren Verlauf wird die Verwendung des Programmes

aus Endanwender-Sicht geschildert.

5.2.3 Benutzung der Software zum aktuellem Stand

Da momentan nur das Rahmenwerk für eine spätere Implementierung von Simulationsansätzen realisiert ist, bietet das Programm nur eine eingeschränkte Funktionalität für den Endnutzer. Auf Erweiterungsmöglichkeiten und dafür vorgesehene Strukturen im Kode wird im Anschluss an dieses Kapitel eingegangen. Sofern eine Verbindung mit der Da-

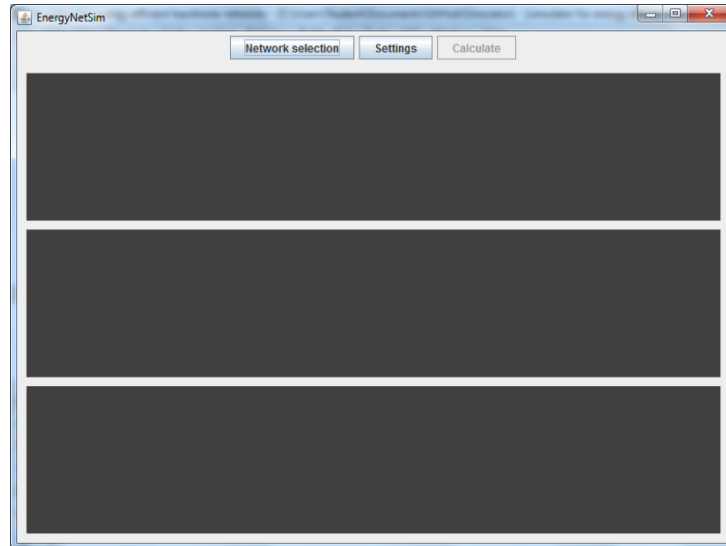


Abbildung 13: Hauptfenster nach dem Starten der Software

tenbank aufgebaut werden konnte, erhält der Benutzer nach Start der Anwendung das Hauptfenster wie in Abbildung 6 angezeigt. Es besteht die Möglichkeit zur Auswahl verschiedener Netze, die in der Datenbank angelegt wurden, über den Menüpunkt „Network selection“ und zur Änderung von Parametern wie beispielsweise dem Strompreis oder dem Netzlastprofil über die Schaltfläche „Settings“. Die geänderten Werte werden in der

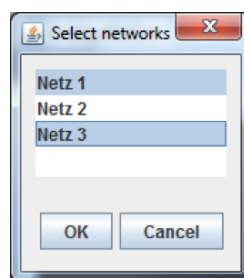


Abbildung 14: Dialoge „Select networks“ und „Settings“

Datenbank hinterlegt, momentan jedoch von der Kalkulationsmethode nicht weiterverwendet. Stattdessen sind in ihr Werte für den Stromverbrauch, die Netzlast und Kosten hinterlegt, deren Ermittlung in Kapitel 5.3 beschrieben wird. Durch Klick auf die Schaltfläche „Calculate“ werden die Werte in Form von Histogrammen visualisiert.

5.2.4 Erweiterungsmöglichkeiten

Die beschriebene Software stellt freilich keine abgeschlossene Lösung zur Wirtschaftlichkeitsbetrachtung von energieeffizienten Konzepten in simulierten Netzen dar. Wie schon

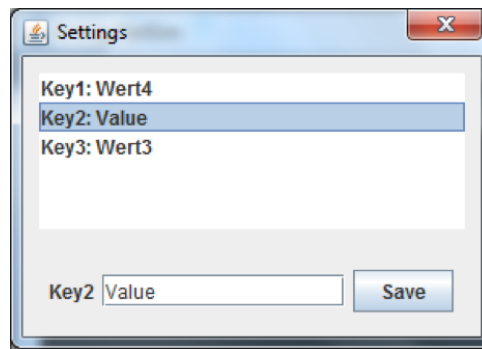


Abbildung 15: Dialoge „Select networks“ und „Settings“

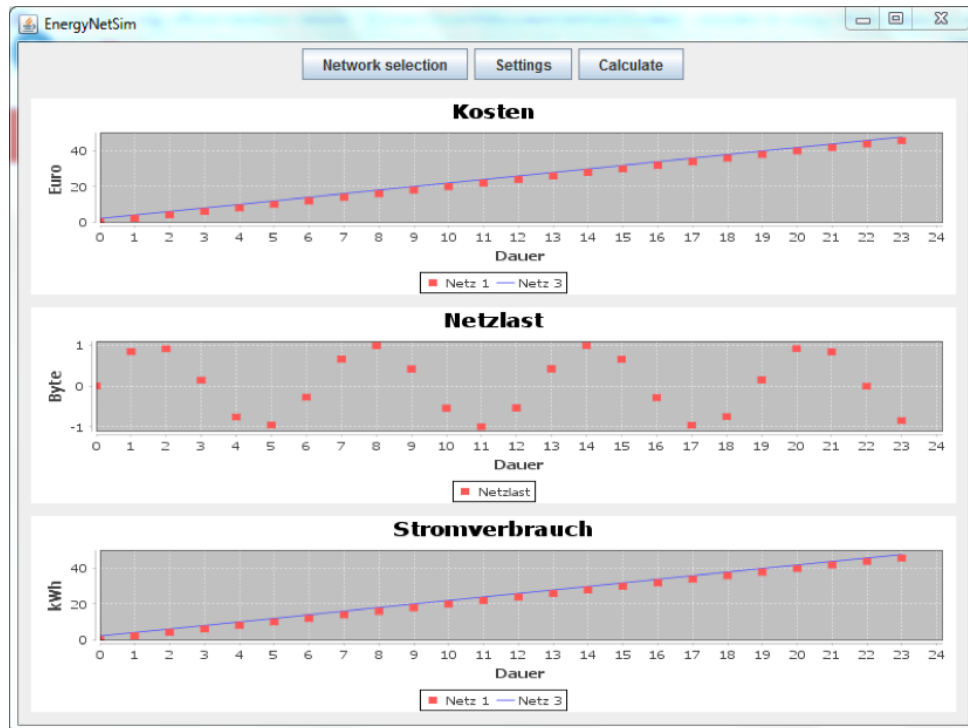


Abbildung 16: Ausgabe der Ergebnisse in Diagrammform

zu Beginn des Kapitels erwähnt, konnte die Entwicklung nicht in der zur Verfügung gestellten Zeit abgeschlossen werden. Vielmehr war die Zielsetzung der Arbeit, eine Anwendung zu erstellen, die es zukünftigen tiefergreifenden Forschungsvorhaben ermöglicht, die fehlenden Algorithmen zu implementieren, welche in Kapitel 5.1 beschrieben sind, ohne auf Aspekte der grafischen Ausgabe sowie der Anbindung an eine externe Datenbank achten zu müssen. Das vorliegende Java-Programm erfüllt diese Anforderungen, indem es folgende vordefinierte Methoden und Datenbankstrukturen enthält, die eine Erweiterung um Simulationsalgorithmen zulassen.

Die Funktion „calculate()“ in der Klasse „MainModel“. Bei Klick auf die Schaltfläche „Calculate“ wird über den Controller in der Klasse „MainModel“ die Funktion „public void calculate()“ aufgerufen. Von dort können der komplette Simulationsalgorithmus gestartet, neue Instanzen von anderen Model-Classen erzeugt und die erhaltenen Ergebnisse in Form einer Liste gespeichert werden.

Das Package „models“. Das Java-Package „models“ bietet Platz für weitere Klassendefinitionen, die von der „calculate()“-Methode der Klasse „MainModel“ aufgerufen

werden und ihrerseits auf die MySQL-Datenbank über die vorhandene Klasse „DatabaseQueries“ zugreifen können. Beispielhaft wurden für die Gerätehardware und die physisch vorhandenen Verbindungen zwischen Knoten die Model-Klassen „HardwareDevices“ und „Link“ angelegt, in denen zukünftig Programmlogik implementiert werden kann.

Das Datenbankschema „energynetsimdb“. Im Rahmen des Software-Engineering-Prozesses wurde ein Datenbankschema entwickelt, das die Datenstruktur für spätere Simulationsalgorithmen abbilden kann. Das zugehörige Entity-Relationship-Diagramm wurde bereits in Kapitel XX vorgestellt und ermöglicht die Generierung der SQL-Create-Befehle für die noch nicht angelegten Relationen.

Zusammenfassend kann festgestellt werden, dass lediglich Änderungen in der Daten- und Logikschicht innerhalb der bereits vorgegebenen Strukturen durchgeführt werden müssen, um den noch nicht kodierten Simulationsalgorithmus einzubinden. Dazu liegt die vollständige Dokumentation der Software dem Anhang bei.

5.3 Abschätzung des Energieverbrauchs

6 Diskussion und Erkenntnisse

Das Ziel des teilweise entwickelten Algorithmus ist das effiziente und realitätsnahe abschätzen der zu erwartenden Energieverbrauchswerte von Backbone-Netzwerken sowohl mit als auch ohne Einsetzen der Energieeinspartetechnologien. Es stellte sich während der genaueren Analyse und zu Anfang der Implementierungsphase der Software allerdings heraus, dass der gewählte generische Ansatz einerseits sehr komplex ist und andererseits auf vielen Annahmen und Verallgemeinerungen beruht, die im Vergleich zu echten existierenden Backbone-Netzwerken die Vergleichbarkeit und damit die Praxistauglichkeit der in Entwicklung befindlichen Softwarelösung fast unmöglich machen. Um die Schwächen des entwickelten Algorithmus auszugleichen oder abzuschwächen ist deutlich mehr Bearbeitungszeit notwendig als für diese wissenschaftliche Arbeit zur Verfügung steht. Weiterhin zeigt eine Begutachtung der Aufgabenstellung dieser Arbeit, dass die detaillierte Ausarbeitung eines solchen Algorithmus nicht Teil des geforderten oder erwartbaren Umfangs ist. Aufgrund der genannten Probleme entschieden wir uns dazu, auf die Dokumentation der geleisteten Entwicklung und die Erfüllung der Aufgabenstellung dieser Arbeit den Schwerpunkt zu setzen.

7 Ausblick

Die diskutierten Ergebnisse machen deutlich, dass für eine vollständige Betrachtung der Wirtschaftlichkeit energie-effizienter Netzkonzepte weitere Forschung notwendig ist. So konnten im Rahmen der vorliegenden Arbeit nur einige Technologien. Hybrid Optical Switching nicht im Algorithmus berücksichtigt

Der durch die Projektgruppe erarbeitete Programmcode kann genutzt werden, um den vorgeschlagenen Algorithmus zur Stromverbrauchs-Berechnung

Um die Wirtschaftlichkeit und damit die Attraktivität energieeffizienter Netzkonzepte für die Netzbetreiber..... Muss CAPEX / Investitionskosten berücksichtigt werden.

Im Sinne der eingangs erwähnten Problemstellung (zurück zu Umweltaspekt).