

Software Defined Networking for RSU Clouds in support of The Internet of Vehicles

Christoph Wittmann

Lehrstuhl für Kommunikationsnetze

Technische Universität München

80333 München

Email: christoph.wittmann@tum.de

Abstract—

I. EINLEITUNG

Intelligente Transport Systeme sind die Zukunft unserer Schienen, Straßen sowie Luft- und Wasserwege und ein großes aktuelles Forschungsgebiet. Durch die damit einhergehende Zusammenführung von Transport und Kommunikation wird sich ein großes Plus in Sachen Sicherheit, Effizienz und Ökologie versprochen. Dies soll unter anderem durch intelligente Ausbalancierung des Verkehrs zur Stauvermeidung oder durch die Warnung vor potentiellen Gefahrenstellen in Echtzeit erreicht werden. Gerade im Straßenverkehr werden jedoch von den Verkehrsteilnehmern auch immer mehr Dienste zur Unterhaltung, wie zum Beispiel Video Streaming, mobiles Internet und Internettelefonie nachgefragt. Diese Nachfrage an bandbreitenhungrigen Anwendungen stellen große Anforderungen an ein Kommunikationsnetz. Hinzu kommt, dass sich der Bedarf an Ressourcen durch die naturgemäß hohe Mobilität der Verkehrsteilnehmer dynamisch ändert. Die Ressourcen müssen in der Regel bei einem Dienstleister angemietet werden und verursachen Kosten für einen Dienstanbieter. Um diese Kosten zu minimieren erfordert dies in der Praxis eine ständige bedarfsgerechte Umverteilung der Ressourcen und eine damit verbundene Neukonfiguration des Kommunikationsnetzes. Einen Lösungsansatz für diese Problemstellung hat Salahuddin et al. in einem Paper mit dem Titel: "Software Defined Networking for RSU Clouds in support of The Internet of Vehicles" vorgestellt. Dort wird eine neuartige Cloud Architektur beschrieben, die durch das Ausnutzen der Vorteile von SDN in hohem Maße flexibel konfigurierbar ist. Konkret bedeutet das, dass innerhalb der Cloud einzelne Dienste verschoben oder an einen anderen Ort kopiert werden, um der wechselnden Nachfrage gerecht zu werden.

Mein eigener Beitrag ist die Zusammenfassung der in diesem Paper vorgestellten Cloud sowie die Ergänzung durch relevante Hintergrundinformationen wie SDN oder die Darstellung anderer Ansätze.

Dieses Paper gliedert sich wie folgt. Im Abschnitt II wird eine kurze Einführung in SDN und dessen Eigenschaften gegeben. In Abschnitt III finden sich eine Darstellung des

Aufbaus einer RSU Cloud und deren Komponenten sowie die Anforderungen, die an eine RSU Cloud gestellt werden. Andere Ansätze und deren Vor- und Nachteile werden in Abschnitt IV erläutert. In Abschnitt V werden das Modell zum Ressourcen Management in der Cloud vorgestellt. In Abschnitt VI werden die Ergebnisse der Simulationen präsentiert und mit anderen Ansätzen verglichen. Abschließend werde ich in Abschnitt VII die Ergebnisse zusammenfassen und einen Ausblick auf weiterführende Arbeit geben.

II. EINFÜHRUNG IN SOFTWARE DEFINED NETWORKING (SDN)

In diesem Kapitel wird ein Überblick über das Themengebiet Software Defined Networking gegeben und die elementaren Eigenschaften des Konzepts dargelegt. Ziel dieses Kapitels ist es, die zum Verständnis notwendigen Hintergrundinformationen zu SDN zusammenzufassen und das Bewusstsein dafür zu schärfen, was SDN eigentlich ausmacht. Um eine Technologie als SDN zu bezeichnen müssen laut [1] vier Prinzipien erfüllt sein, die ich im Folgenden kurz zusammenfassen möchte. Der zentrale Grundsatz in SDN ist die Trennung des Netzwerks in eine physikalische Datenebene und eine davon abstrahierte Kontrollebene. Das bedeutet, dass in einer SDN-fähigen Netzwerkkomponente eine von der Datenebene abgekoppelte, externe Instanz existieren muss. Diese Instanz wird Controller genannt und zeichnet sich dadurch aus, dass sie die Möglichkeit hat, das Weiterleitungsverhalten einer Netzwerkkomponente direkt zu bestimmen. In der Praxis besteht ein SDN daher aus Controllern und Switches. In einem Switch sind nur noch die, von ihm auszuführenden Weiterleitungsvorschriften gespeichert, die dynamisch von einem Controller über die Kontrollebene aktualisiert werden können. Daraus ergibt sich zum Beispiel der Vorteil, dass Daten- und Kontrollebene im Design beziehungsweise in Untersuchungen und Analysen getrennt voneinander betrachtet werden können. Ein weiteres Prinzip ist, dass ein derartiger Controller, obwohl er sich in einem SDN Netzwerk auf mehrere virtuelle oder physikalische Bestandteile verteilen kann, sich trotzdem logisch wie eine zentrale Einheit verhält, die über globales Wissen verfügt. Diese Eigenschaft bringt den Vorteil der schnelleren und effizienteren dynamischen Anpassung des Netzwerks. Als dritten Grundsatz wird in [1] die Offenheit der Schnittstellen genannt, worauf ich hier aber aus Relevanzgründen nicht näher eingehen möchte. Viel interessanter ist die Eigenschaft der Programmierbarkeit eines SDN Netzes. Dies wird durch die Entkopplung von Daten- und

*This paper is a reinterpretation of the paper M.A. Salahuddin. "Software Defined Networking for RSU Clouds in support of The Internet of Vehicles," *Internet of Things Journal*, IEEE, 2014, pp. 133-144. It was presented on June 22, 2015 (Paper submission deadline) as a part of MSCE Seminar (MSCE course TUM), under the supervision of M. Sc. Christian Sieber (c.sieber@tum.de).

Kontrollebene ermöglicht und macht das Netzwerk extrem anpassbar an verschiedene Anwendungen. Laut [1] lässt sich das Netzwerk dann als eine einzige programmierbare Einheit behandeln und nicht als eine Anhäufung von Geräten, die individuell konfiguriert werden müssen.

OpenFlow ist das am weitesten verbreitete Kommunikationsprotokoll, das in SDN eingesetzt wird. Es fungiert als Schnittstelle zwischen Datenebene und Kontrollebene, indem es direkten Zugriff auf Weiterleitungsverhalten eines Switches ermöglicht. Man spricht daher auch häufig von OpenFlow Controllern oder OpenFlow Switches.

III. ARCHITEKTUR DER RSU CLOUD

Ein Fahrzeugnetzwerk (Vehicular Grid) besteht grob aus zwei Arten von Komponenten. Auf der einen Seite sind das Lokalisierungssysteme, Sensoren, Mikroprozessoren und Funkeinheiten, die an einem Fahrzeug montiert sind - sogenannte On-Board Units (OBUs) - und auf der anderen Seite Sensoren und Mikrocontroller, die fest in die Straßen integriert sind, genannt Roadside Units (RSUs). Diese Komponenten vernetzt bilden ein sogenanntes Fahrzeug Ad hoc Netzwerk (VANET). Der neue Ansatz, der in diesem Paper diskutiert wird, ist die Zusammenfassung der RSUs zu einer RSU Cloud. Da eine derartige Cloud demnach alle nachgefragten Sicherheits- sowie Unterhaltungsdienste bereitstellen können muss, werden hohe Anforderungen an sie gestellt. Eine RSU Cloud muss nicht nur verlässlich sondern auch sehr leistungsstark in Sachen Datenrate und Latenz sein, um größtmögliche Sicherheit zu garantieren und somit die Akzeptanz in der Bevölkerung zu erlangen.

Eine solche RSU Cloud besteht dabei neben gewöhnlichen RSUs zudem aus RSU Mikrodatenzentren. Ein Mikrodatenzentrum besteht aus einer kleinen Recheneinheit und einem OpenFlow Switch, sodass es in ein SDN Netzwerk eingebunden werden kann. Softwareseitig besteht ein Mikrodatenzentrum aus einem Betriebssystem und einem Hypervisor, der es mittels Virtualisierung ermöglicht mehrere virtuelle Maschinen dynamisch auf einem Host auszuführen. Auf einem solchen Mikrodatenzentrum sollen dann die Dienste gehostet werden, die von den Verkehrsteilnehmern nachgefragt werden. Dies hat den großen Vorteil, dass die Dienste sehr "nah" an den Konsumenten sind, was sich in einer höheren Verlässlichkeit und kürzeren Latenzzeiten bemerkbar macht. Die, zum Aufspannen einer Cloud in einem SDN Netzwerk, nötigen Softwarekomponenten wie ein OpenFlow Controller oder ein Cloud Controller sollen ebenfalls in manchen Mikrodatenzentren ausgeführt werden. Eine weitere neue Komponente, die im Originalpaper vorgestellt wurde, ist der RSU Cloud Resource Manager (CRM). Der CRM ist ebenfalls in manchen Mikrodatenzentren integriert und soll Informationen über das Hosten von Diensten, die Verschiebung von Diensten oder das Ändern von Weiterleitungsvorschriften verbreiten, indem er mit OpenFlow und Cloud Controllern kommuniziert. Des Weiteren hat er die Aufgabe, die richtigen Entscheidungen bezüglich der optimalen Lage und Anzahl an gehosteten Diensten und den dazugehörigen Weiterleitungsvorschriften zu treffen. Entscheidend ist hierbei, dass die Anzahl an Verschiebungen von virtuellen Maschinen möglichst gering gehalten wird, da sie zusätzlichen Netzwerkverkehr in Daten und Kontrollebene produzieren und somit negativen Einfluss auf Quality of Service und darüber hinaus auch zusätzliche Kosten für den Anbieter

eines Dienstes haben. Im Weiteren soll nun die Formalisierung des zusätzlichen Netzwerkverkehrs, der durch eine Neukonfiguration des Netzes entsteht, beschrieben werden. Zunächst einmal ist der Begriff der Konfiguration einzuführen, der den Zustand des Netzes zum Zeitpunkt t_i beschreibt. Eine Konfiguration ist somit definiert als 3-Tupel $\langle X^{t_i}, Y^{t_i}, Z^{t_i} \rangle$ mit den Komponenten X^{t_i} welche den Satz an Hosts darstellt, Y^{t_i} die den Satz an Weiterleitungsvorschriften repräsentiert und Z^{t_i} , dem Satz an Vorschriften für die Weiterleitung eines Eingangs an mehrere Empfänger, den sogenannten Gruppenvorschriften. Zur Bestimmung des verursachten Netzwerkverkehrs in der Datenebene werden nun die Verschiebungen von Diensten gezählt. Formal kann man die Verschiebungen vom Zeitpunkt t_i zum Zeitpunkt t_{i+1} mit der Formel $|X^{t_i} - X^{t_{i+1}}|$ ausdrücken. Ein Spezialfall ist hier das Löschen eines Dienstes von einem Host, was nicht berücksichtigt wird, da kein Datenaustausch auf der Datenebene erforderlich ist. Für die Analyse des Verkehrs in der Kontrollebene werden die Veränderungen der Weiterleitungs- und Gruppenvorschriften herangezogen. Der Verkehr setzt sich hier aus der Summe der Weiterleitungsbeziehungsweise Gruppenvorschriften, die hinzugefügt oder gelöscht werden und außerdem aus der Anzahl der Weiterleitungsvorschriften die zu Gruppenvorschriften und der Gruppenregeln die zu Weiterleitungsvorschriften werden, zusammen. Daraus ergibt sich folgende Formel für den zusätzlichen Verkehr in der Kontrollebene: $|Y^{t_i} - Y^{t_{i+1}}| + |Y^{t_{i+1}} - Y^{t_i}| + |Z^{t_i} - Z^{t_{i+1}}| + |Z^{t_{i+1}} - Z^{t_i}| + |Y^{t_i} \cap Z^{t_{i+1}}| + |Z^{t_i} \cap Y^{t_{i+1}}|$.

IV. "RELATED WORK" ODER HINTERGRUND(SIEHE III. BACKGROUND)

- Darlegung anderer Ansätze und deren Vor- und Nachteile
 - Cloud Computing in VANETs
 - Cloud Resource Management

V. CLOUD RESOURCE MANAGEMENT

A. ILP Ansatz

B. Heuristischer Ansatz

1) Verbesserung durch Reinforcement Learning:

VI. SIMULATIONSERGEBNISSE

- Vergleich zwischen drei verschiedenen Ansätzen
- Verbesserung der Heuristik durch Reinforcement Learning

VII. FAZIT

- Zusammenfassung der Fakten
- Weiterführende Arbeit

DANKSAGUNG

Notwendig/üblich?

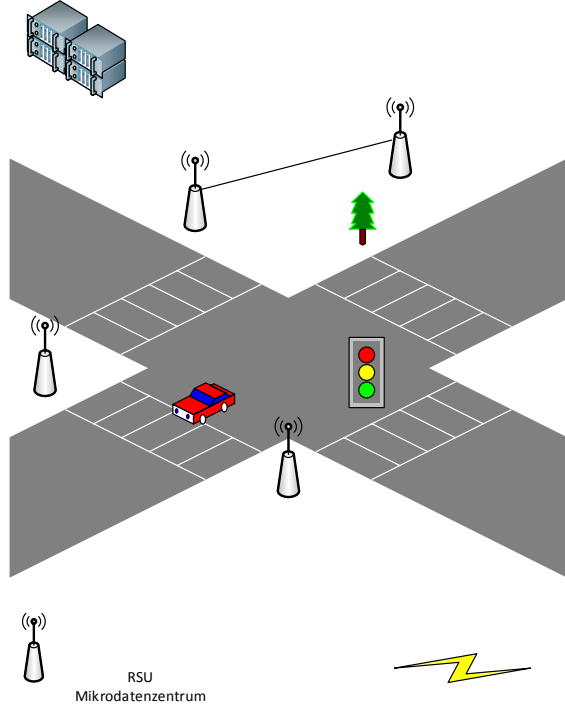


Fig. 1. eine Grafik ohne Sinn und Verstand

REFERENCES

- [1] Michael Jarschel, Thomas Zinner, Tobias Hoßfeld, Phuoc Tran-Gia, and Wolfgang Kellerer *Interfaces, attributes, and use cases: A compass for SDN*, Communications Magazine, IEEE, 2014.
- [2] Mohammad A. Salahuddin, Ala Al-Fuqaha, Mohsen Guizani *Software Defined Networking for RSU Clouds in support of The Internet of Vehicles*, IEEE Internet of Things Journal, VOL. X, NO. X, November 2014.
- [3] Mohammad A. Salahuddin, Ala Al-Fuqaha, Mohsen Guizani and Soumaya Cherkaoui *RSU Cloud and its Resource Management in support of Enhanced Vehicular Applications*, Globecom 2014 Workshop - Cloud Computing Systems, Networks, and Applications, 2014.

VIII. FRAGEN/UNKLARHEITEN

$$\min \left\{ \sum_{m=1}^N \sum_{n=1}^N \sum_{x=1}^{k_{m,n}} \sum_{k=1}^S (1 - \rho) \cdot (\beta_k^{m,n,x} + \gamma_k^{m,n,x}) + \sum_{m=1}^N \sum_{k=1}^S \rho \cdot \alpha_{m,k} \right\} \quad (1)$$

$$\min \left\{ \sum_{m=1}^N \sum_{k=1}^S P \cdot h_{m,k} + \sum_{e=0}^{|E|} (1 - P) \cdot \frac{d_e}{q_{c_e,e}} \right\} \quad (2)$$