

Implementation von DSR

Onur Kilic

Telematik-Projekt: Wireless Embedded Systems

Institute of Computer Science

Freie Universität Berlin, Germany

kilic@zedat.fu-berlin.de

Zusammenfassung—In diesem Bericht geht es um die Implementierung von Routing Protokoll DSR. Es wird gezeigt, wie das Protokoll aufgebaut ist und wie es funktioniert. Die Implementierung die in Rahmen des Projekts gemacht wurde, ist ein abgeschwächte Form von DSR. Es wird gezeigt, was nicht Implementiert wurde. Außerdem wurde diese Implementierung getestet und die Ergebnisse von diesem Tests wurde in Form von Tabelle und Diagramme beschrieben.

I. INTRODUCTION

In diesem Projekt geht es um Implementierung eines MANET- Routingprotokolls. Genau gesagt, geht es um die Implementierung von DSR. Ein **mobiles Ad-hoc-Netzwerk (MANET)** ist ein Funknetz. In diesem Netz gibt es zwei oder mehrere Stationen, die ein temporäres Netzwerk ohne vorhandene Infrastruktur aufbauen können und keine zentrale Verwaltung besitzen.

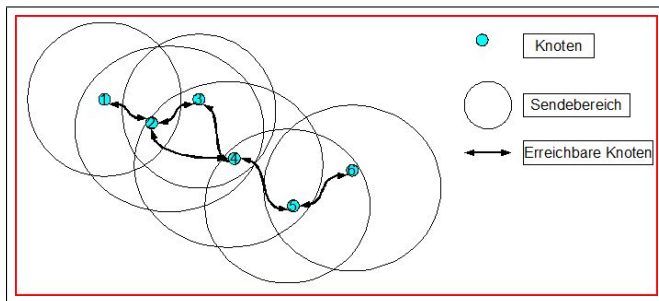


Abbildung 1. Abbildung 1.0 (Manet)

Also besitzen Ad-hoc-Netzwerke keinen festen Basisstationen und jede Station ist gleichzeitig ein Endstation und ein Router. Die Abbildung 1.0 zeigt auch dieses Merkmal, da hat jeder Knoten eine Sende- Empfangsbereich. Also z.B. der Knoten 1 kann nur zu Knoten 2 etwas schicken oder von Knoten 2 etwas empfangen. Bei mobile Ad-hoc-Netzwerke sind die Knoten in Bewegung. In mobile Ad-hoc-Netzwerke kann es häufige Topologiewechsel zu Stande kommen. Außerdem haben die Knoten geringere Leistung (Batterie) und haben geringere Bandbreite, deswegen muss es neue Routing

Protokolle geben, die mit diesen Eigenschaften gut arbeiten. Ad-hoc Routing Protokolle kann man in drei Gruppen zusammenfassen. Erste Gruppe ist Proaktiv wie OLSR, zweite Reaktiv wie DSR, AODV und dritte Hybrid wie ZRP. Routing findet auf Ebene 3 des ISO/OSI-Referenzmodells statt und Hauptaufgabe des Routings ist, Datenpakete von der Quelle zum Ziel weiterzuleiten.

In section II Basis von DSR. Detaillierte Information über die Implementierung section III. Die Zusammenfassung section IV.

II. BASIC

A. Dynamic Source Routing (DSR)

Wie in der Einleitung steht, ist DSR ein reaktive Protokoll, deswegen ist es gut, dass man beschreibt, was mit reaktive Protokolle gemeint sind. Die reaktiven Routing-Protokolle suchen einen Weg durch das Netz erst, wenn dieser benötigt wird. D.h. beim ersten Senden eines Paketes von einem Knoten zu einem anderen wird der entsprechende Weg ermittelt. Dies hat den Vorteil, dass keine ungenutzten Wege berechnet werden, was den Overhead der Routing-Protokolle reduziert. Allerdings hat dieses Verfahren den Nachteil, dass das erste Paket mit einer erhöhten Verzögerung eintrifft, da zunächst der Weg ausgewählt werden muss.

DSR basiert auf dem Link-State-Algorithmus. Bei den Link-State Routingprotokollen wird von jedem Knoten erwartet, dass er zumindest einen Teil des gesamten Netzwerkes kennt, die Kenntnis des gesamten Netzes ist jedoch nicht zwingend nötig. Ändert sich eine Verbindung, also der Link-State, so wird dies an das Netzwerk per Flooding, also dem Zustellen an alle erreichbare Nachbarn, mitgeteilt. Daraufhin passen alle Knoten ihre Linktabelle an.

Die Haupteigenschaften von DSR Protokoll sind:

- Vorhandene Routeninformationen so lange wie möglich nutzen
- Jeder Knoten hat seine eigene Sicht auf das Netz und seine eigenen Routen

- Es können mehrere Routen zu einem Ziel abgelegt werden

Wenn ein Knoten etwas senden will, dann schaut er in seinem Route Cache nach, ob bereits eine Route für den Zielknoten vorliegt. Wenn so eine Route existiert, dann sendet er über diese Route (Source Route). Aber wenn so eine Route in seinem Route Cache nicht existiert, dann startet er Route Discovery.

Route Discovery

Bei Route Discovery sendet der Knoten per Broadcast zu allen Knoten, die er Erreichen kann ein Request-Paket. Dabei schickt der Knoten die Zieladresse und eine Liste, wo er auch seine Id geschrieben hat, zu diesen erreichbaren Knoten. Diese Knoten vergleichen die Zieladresse mit den eigenen Adressen, wenn die Adressen nicht gleich sind, schicken sie weiter. Aber wenn die Zieladresse mit der eigenen Adresse gleich ist, dann sendet der Knoten ein Reply-Paket. Der Zielknoten schickt den Reply-Paket per Source Routing. Die Abbildung 1.1 zeigt ungefähr wie Route Discovery funktioniert.

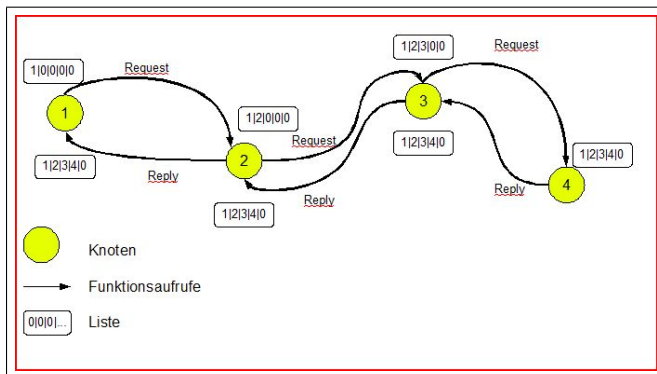


Abbildung 2. Abbildung 2.0 (Route Discovery)

Route Maintenance

Die Aufgaben von Route Maintenance sind:

- Stellt fest, wenn sich die Netzwerktopologie ändert, z.B., wenn ein Zwischenknoten, der in einer Wegewahl vorgesehen ist, plötzlich ausfällt
- In diesem Fall trifft die Route Maintenance die Entscheidung, ob eventuell eine alternative Route verwendet wird (falls diese vorhanden ist) oder ob eine neue Route bestimmt werden soll
- Letzteres initiiert die Route Discovery

Die Abbildung 2.1 verdeutlicht diesen Sachverhalt.

Zu Beachten ist Route Discovery und Route Maintenance reagieren nur auf Anfrage.

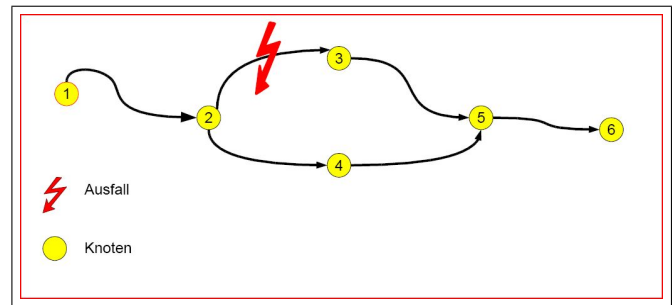


Abbildung 3. Abbildung 2.1 (Route Maintenance)

Route Caching

Mit Route Caching wird DSR optimiert. Die Abbildung 2.2 zeigt, wie die Route Caching funktioniert.

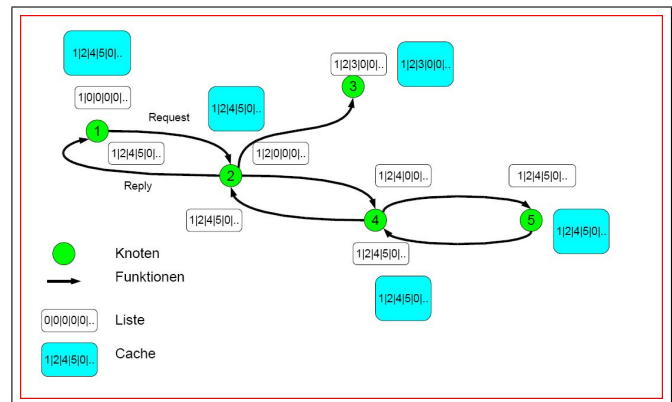


Abbildung 4. Abbildung 2.2 (Route Caching)

Nach dem Request-Paket von dem Startknoten 1 gesendet wurde, kommt dieser Paket auch zu Knoten 3, obwohl dieser Knoten nicht zum Ziel bringt, lernt dieser Knoten, dass sein Nachbar Knoten 2 ist und außerdem über diesen Knoten Knoten 1 erreichen kann. Die anderen Knoten die zwischen Startknoten und Zielknoten liegen lernen auch. Ein Beispiel: Knoten 2 weißt durch Route Caching, dass dieser Knoten über Knoten 4 Knoten 5 erreichen kann. Wenn z.B. Knoten 2 zu Knoten 5 etwas senden will, dann braucht dieser Knoten kein Route Discovery, was dadurch Zeit und Leistung gespart wird.

DSR hat sowie Vorteile und Nachteile.

DSR Vorteile

- Nur Routen zwischen Knoten, die miteinander kommunizieren (Reduziert den Overhead von Route Maintenance)
- Route Caching kann zusätzlich den Route Discovery Overhead reduzieren

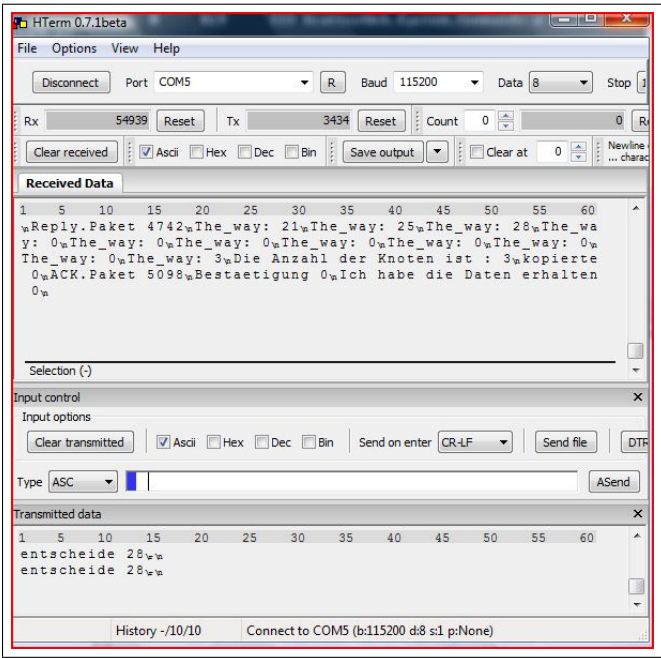


Abbildung 7. Abbildung 3.2 (Ausgabe von der Quelleknoten)

die Paket- Typen und die Bestätigung, dass Zielknoten die Daten erhalten hat.

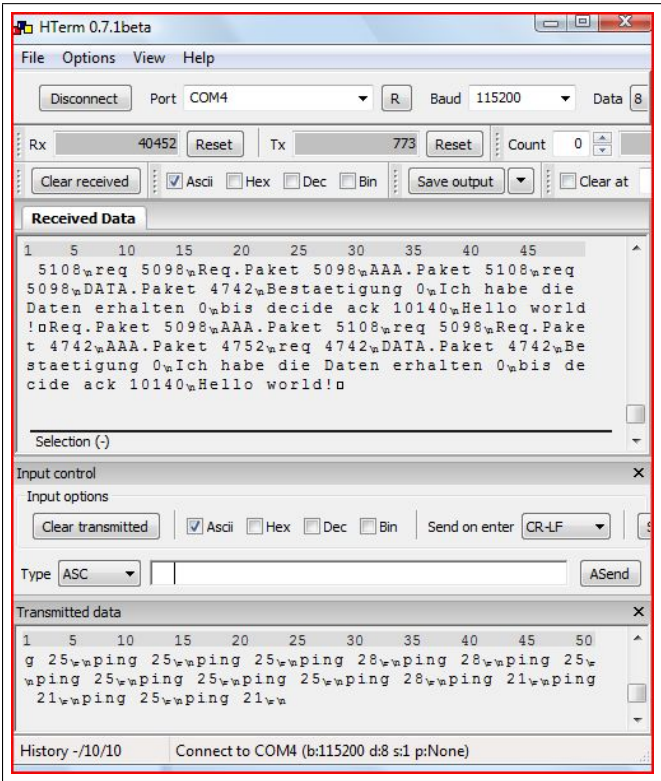


Abbildung 8. Abbildung 3.3 (Ausgabe von der Zielknoten)

Die Abbildung 3.3 zeigt, was der Zielknoten ausgibt. Zielknoten gibt die Paket- Typen und den Payload aus.

B. Test

Der Test besteht aus zwei Teilen. Erste Teil gibt an, wie viel von 100 Pakete den Ziel erreichen und der zweite Teil, wie groß darf der Payload sein. Als Topologie habe ich Bus Topologie gewählt und die Anzahl der Knoten ist 3.

1.Teil

Die Anzahl der Knoten war bei der Test 3, weil andere Knoten nicht arbeiteten. Am Anfang war ich sehr schockiert, weil ich ein Ergebnis z.B von 100 Paketen nur 23 Daten- Pakete(angekommene Daten) hatte. Aber nach der Wechsel der Batterie habe ich erkannt, dass der Grund die Batterien war. Nach dem Wechsel waren die Ergebnisse sehr gut. Ich habe dadurch gelernt, dass man vor einem Test die Knoten und deren Batterien überprüfen soll, damit man nicht zu falschen Ergebnisse kommt, dass mit Implementierung nicht zutun hat. Ich habe 100 Pakete pro Test geschickt und habe beobachtet, wie viele davon zu den Zielknoten kamm und wie viel davon bestätigt wurde. Die Tabelle 1.0 zeigt die Ergebnisse.

Test	angekommene Daten	Bestätigte Daten
1	81	65
2	87	62
3	79	60
4	83	64
5	82	62
6	85	67
7	89	64
8	80	62
9	85	62
10	82	63

Tabelle 1.0:Ergebnisse von Test

Außerdem habe ich davon ein Signalendiagramm gemacht (siehe Abbildung 3.4) Nach meiner Meinung sind die Ergebnisse in Ordnung, also denke ich, dass die Implementierung gut geklappt hat. Von 100 Pakete sind ungefähr durchschnittlich 83 Pakete den Ziel erreicht und davon wurde durchschnittlich 62 Pakete bestätigt.

2.Teil

In der zweiten Teil habe ich nur zwei Knoten genommen. Ich habe zuerst 50 Byte Payload geschickt, also etwas geschrieben, was 50 Byte entsprach. Da konnte ich beobachten, dass die Daten den Ziel erreicht hatten. Aber die Daten waren nicht vollständig, deswegen habe

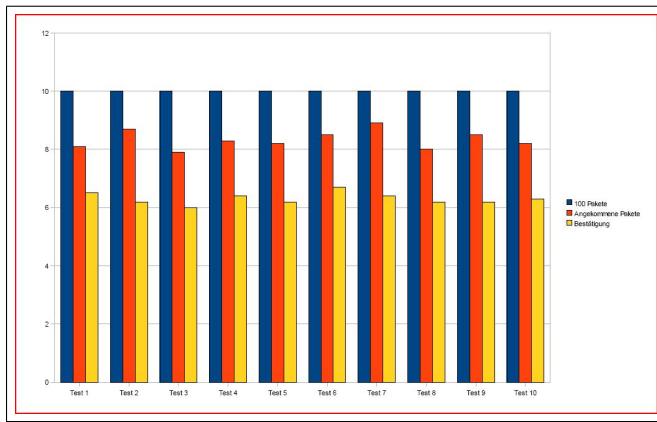


Abbildung 9. Abbildung 3.4 (Diagramm von Test)

ich kleine als 50 Byte geschickt und durch mehrmalige Versuche bin ich zu dem Ergebniss gekommen, dass mehr als 46 Byte nicht richtig gesendet wird. In diesem Fall habe ich nur mit zwei Knoten dieser Test gemacht, weil was mit zwei Knoten nicht klappt, würde mehr als zwei Knoten sowieso nicht funktionieren.

IV. CONCLUSION

DSR ist ein reaktive Protokoll und hat Vorteile und Nachteile. Eine der wichtigste Vorteil ist, dass DSR Strom- und Bandbreitesparsam ist. So eine Eigenschaft ist bei MANET sehr wichtig, da z.B. die Knoten mit Batterie arbeiten und haben auSSerdem eine geringere Bandbreite. Aber die Eigenschaft, der Paket Header wächst mit der Weglänge, kann ein Problem in groSSen Netzwerke mit vielen Knoten sein. Die Implementierung von DSR sah leicht aus, aber man merkte schnell, dass die Implementierung nicht einfach ist, obwohl nicht alles (z.B. Route Cache) Implementiert wurde. AuSSerdem wurde die Implementierung getestet und der Test zeigt gut die Grenzen der Implementierung.

LITERATUR

- [01] ndrew S. Tanenbaum Computernetzwerke Pearson Studium 2003
- [02] [ttp://de.wikipedia.org/wiki/Mobiles_Ad-hoc-Netz](http://de.wikipedia.org/wiki/Mobiles_Ad-hoc-Netz)
- [03] [ttp://casablanca.informatik.hu-berlin.de/database/ns-2-PS/VL/draft-ietf-manet-dsr-10.pdf](http://casablanca.informatik.hu-berlin.de/database/ns-2-PS/VL/draft-ietf-manet-dsr-10.pdf)
- [04] [ttp://web.informatik.uni-bonn.de/IV/martini/Lehre/Veranstaltungen/WS0405/Sem_Rechnernetze/Vortraege/Vortrag_AndreasBenden.pdf](http://web.informatik.uni-bonn.de/IV/martini/Lehre/Veranstaltungen/WS0405/Sem_Rechnernetze/Vortraege/Vortrag_AndreasBenden.pdf)

LITERATUR