

Ameisen-Routing-Algorithmus(ARA) für MANET

Yu Weng, Hans-Christoph Schemmel
Freie Universität Berlin
{weng,schemmel}@inf.fu-berlin.de

Zusammenfassung—Mit Hilfe des Ant-Routing-Algorithmus (ARA) ist es möglich, mehrere Knoten drahtlos miteinander kommunizieren zu lassen, wobei es sich bei ARA um ein sogenanntes *reaktives Routingverfahren* handelt. Der Anwendungsbereich von ARA liegt dabei vor allem bei Mobilien Ad-hoc Netzen. Diese Art von Netzen benötigen keine vorhandene Infrastruktur und finden in vielen Bereichen Einsatz, wie etwa bei Netzen aus Sensorknoten. Solche Knoten sind über ein größeres Areal verteilt und tauschen Informationen untereinander aus, wie etwa Temperatur oder seismologische Aktivitäten, ohne die genaue Topologie zu kennen. Aufgrund dieser Bedingungen werden an die verwendeten Routing-Protokolle besondere Anforderungen gestellt.

In diesem Bericht sollen die Funktionsweise und Besonderheiten von ARA erläutert und untersucht werden. Dazu wird in den ersten beiden Kapiteln das Protokoll und die entwickelte Implementierung diskutiert. Im Anschluss daran, werden die durchgeführten Tests zur Evaluation des Protokolls und der Implementierung beschrieben und die erlangten Ergebnisse ausgewertet. Abschließend gibt es ein Fazit zum ARA-Protokoll.

Index Terms—MANET, ARA, Ameise-Routing-Algorithmus.

I. ANT-ROUTING-ALGORITHMUS

A. Das Futtersuchverhalten von Ameisen

Die grundlegende Idee vom Ameisen-Algorithmus ist aus dem Futtersuchverhalten von realen Ameisen abgeleitet.

Die Ameisen gehen dabei von ihrem Nest aus in unterschiedliche Richtungen um Futter zu finden. Wenn sie nun auf Wegverzweigungen treffen, müssen sie sich für einen Weg entscheiden. Unterwegs legen die Ameisen eine Pheromonspur auf ihrem genommenen Pfad ab, wodurch dieser Weg markiert wird. Je dichter nun die Pheromonspur ist, desto mehr Ameisen gab es, die diesen Weg genommen haben. Die Dichte bzw. Stärke der Pheromonspur verringert sich mit der Zeit allerdings von alleine.

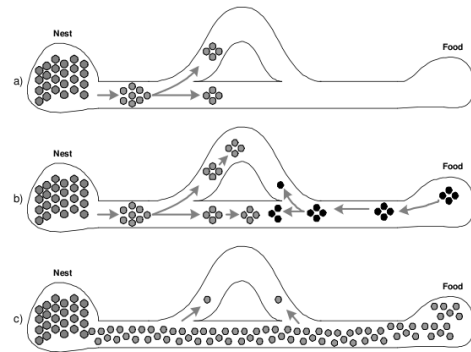


Abbildung 1. Ameisen suchen nach Essen (vgl. [1])

Abbildung (I-A) zeigt, dass es zwei Wege vom

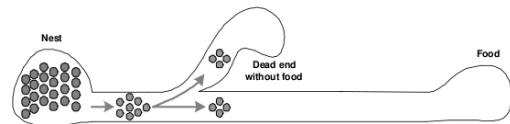


Abbildung 2. Beispiel mit einer Sackgasse (vgl. [1])

Nest zum Futter gibt. Am Anfang wählen die Ameisen zufällig einen Weg aus und erreichen das Essen - siehe dazu a). Auf dem Weg zurück zum Nest werden sie eine zweite Entscheidung treffen müssen. Wenn angenommen wird, dass die Ameisen auf beiden Wegen gleichverteilt sind, dann gibt es im gleichen Zeitraum mehr Ameisen, die den kürzeren Weg gegangen sind als den Längeren. Deswegen hat der kürzere Pfad eine dichtere Pheromonspur und die Ameisen (auf dem Zurückweg) werden sich für ihn entscheiden, wie in Abbildung (I-A) b) zu sehen ist. Die Dichte der Pheromonspur wird dadurch noch schneller erhöht, weshalb fast alle Ameisen am Ende diesen Pfad benutzen werden.

Es kann auch dazu kommen, dass es eine Sackgasse

ohne Essen am Ende gibt, die näher beim Nest liegt (siehe Abbildung (I-A)). Die Ameisen werden trotzdem den Weg zum Essen finden, da Ameisen mit Essen mehr Pheromone legen und sich somit der Pfad zum Essen stärker ausprägt.

B. ARA im MANET

In diesem Abschnitt wird die Anwendung des Ameisen-Algorithmus in einem MANET (*Mobile Multi-Hop Ad-Hoc Network*) diskutiert.

Dabei gibt es drei Phasen im Routing-Algorithmus.

1) Pfadentdeckungsphase: In der Pfadentdeckungsphase werden neue Pfade erzeugt, dies geschieht mit Hilfe von zwei kleinen *Ameisen – Paketen*. Diese sind FANTs (Forward Ant) und BANTs (Backward Ant). Ein FANT wird vom Quellknoten gesendet um einen Pfad zu einem bestimmten Zielknoten zu finden. Dabei wird allerdings der Pheromonwert vom Pfad, der zurück zum Quellknoten führt, verstärkt bzw. ausgeprägt. Analog dazu, wird ein vom Zielknoten gesendetes BANT-Paket behandelt.

Wenn ein Knoten ein Daten-Paket versenden möchte, und keinen entsprechenden Routingtabelleneintrag zum Zielknoten finden kann, dann sendet er ein FANT an all seine Nachbarknoten (Broadcast). Die Nachbarknoten bzw. Zwischenknoten werden es wiederrum an all ihre Nachbarknoten weiterleiten. Dies geschieht solange, bis der Zielknoten das FANT bekommen hat. Daraufhin sendet er ein BANT-Paket an all seine Nachbarn (wiederum Broadcast) zurück. Nachdem der Quellknoten ein BANT erhalten hat, besitzt er mindestens einen Pfad (Routing-Tabellen-Eintrag) zum Zielknoten. Damit kann er sein Datenpaket abschicken.

Jedes FANT/BANT eines Knotens hat dabei eine eindeutige Sequenznummer. Damit sind die Zwischenknoten in der Lage, duplizierte FANT/BANT-Pakete zu erkennen. Ein Knoten, der ein FANT/BANT bekommt, erzeugt ein Eintrag in seiner Routing-Tabelle. Ein solcher Eintrag entspricht dabei einem Tripel der folgende Art: (Zieladresse, nächster Hop, Pheromonenwert). Der Knoten interpretiert die Quelladresse (Absender) eines FANTs/BANTs als Zieladresse. Die Adresse des Knotens, von dem er das Paket erhalten

hat, als den nächsten Hop und errechnet den Pheromonenwert in Abhängigkeit von den Hops, die das FANT/BANT benötigt hat, um diesen Knoten zu erreichen. Anschließend speichert er sich die Sequenznummer und leitet das FANT/BANT an seine Nachbarknoten weiter. Ein Knoten kann ein dupliziertes Paket anhand der Sequenznummer erkennen, solch ein Paket wird normal verarbeitet, jedoch nicht an andere Knoten weitergeleitet.

In Abbildung (I-B1) ist die Pfadentdeckungsphase

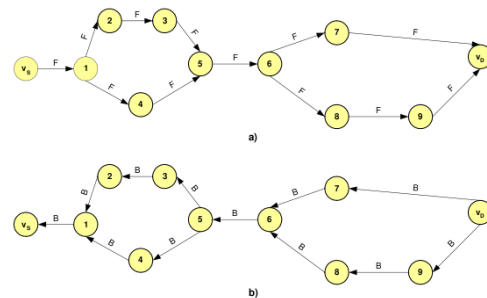


Abbildung 3. Beispiel für den Wegeaufbau mittels FANTs (a) und BANTs (b) (vgl. [1])

von ARA gezeigt. Bild (I-B1) a) zeigt die Ausprägung des Rückpfades mittels FANT, der zum Quellknoten bzw. FANT-Absender führt. Dagegen ist in Abbildung (I-B1) b) zu erkennen, wie ein Weg zum Zielknoten mittels BANTs erzeugt wird. Zu beachten ist dabei, dass Knoten 6 in a) und Knoten 5 in b) mehrere Pfade/Links zum nächsten Knoten besitzen. Dies zeigt, dass ARA Multipaths erzeugt und auch unterstützt. Allerdings können dadurch Schleifen auf der Route beim Versenden von Paketen auftreten. Auf diese Eigenschaft wird im folgenden Abschnitt näher eingegangen.

2) Pfaderhaltungsphase: Nach der ersten Phase können die neu gefundenen Pfade zum Paketversand benutzt werden. Die Routing-Tabelle jedes Knotens enthält die notwendigen Weg-Informationen. Durch jede Nutzung eines Pfades wird sein Pheromonwert erhöht, d.h. wenn ein Knoten einen Eintrag (v_D, v_i, φ) aus seiner Routing-Tabelle verwendet und wenn er ein Paket von einem Knoten erhält, dann wird der jeweilige Pheromonwert φ um $\Delta\varphi$ inkrementiert. Die folgenden Tabellen (I bis IV) zeigen ein Beispiel, in dem Knoten 5 ein Datenpaket - zum Zielknoten (v_D) - an Knoten 6 weiterleitet. In den Tabellen, sind die Routingtabelle-Einträge vor

und nach dem Versenden dargestellt.

Tabelle I
ROUTINGTABELLE VON KNOTEN 5, VOR DEM SENDEN

Zielknoten	Next-Hop	Pheromonwert
v_S	3	φ_1
v_S	4	φ_2
v_D	6	φ_3

Tabelle II
ROUTINGTABELLE VON KNOTEN 6, VOR DEM PAKETERHALT

Zielknoten	Next-Hop	Pheromonwert
v_S	5	φ_4
v_S	7	φ_5
v_D	8	φ_6

Tabelle III
ROUTINGTABELLE VON KNOTEN 5, NACH DEM VERSENDEN DER DATEN

Zielknoten	Next-Hop	Pheromonwert
v_S	3	φ_1
v_S	4	φ_2
v_D	6	$\varphi_3 + \Delta\varphi$

Tabelle IV
ROUTINGTABELLE VON KNOTEN 6, NACH DEM VERSENDEN DER DATEN

Zielknoten	Next-Hop	Pheromonwert
v_S	5	$\varphi_4 + \Delta\varphi$
v_S	7	φ_5
v_D	8	φ_6

Zudem wird der Pheromonwert auch regelmäßig (in r Sekunden) mit einem Faktor $(1 - q)$ multipliziert. Die Tabellen V und VI zeigen dieses Dekrementierungsverfahren.

Um eine Endlosschleife zu vermeiden - welche durch Multipaths entstehen können - verwendet ARA ein spezielles Flag, welches *Flag_Duplicate* heißt. Die Knoten können mit Hilfe der Sequenznummer und dem Quellknoten vom Datenpaket doppelt erhaltene Pakete identifizieren. Falls ein Knoten ein dupliziertes Datenpaket bekommt, wird er dieses Flag setzen und das Paket zurück zum Vorgänger senden. Der Vorgänger wird den entsprechenden Eintrag/Link zum Knoten in seiner Routing-Tabelle löschen. Anschließend wird er einen neuen Weg herausuchen.

Tabelle V
ROUTINGTABELLE VON KNOTEN 5 NACH r SEKUNDEN

Zielknoten	Next-Hop	Pheromonwert
v_S	3	$\varphi_1 * (1 - q)$
v_S	4	$\varphi_2 * (1 - q)$
v_D	6	$(\varphi_3 + \Delta\varphi) * (1 - q)$

Tabelle VI
ROUTINGTABELLE VON KNOTEN 6 NACH r SEKUNDEN

Zielknoten	Next-Hop	Pheromonwert
v_S	5	$(\varphi_4 + \Delta\varphi) * (1 - q)$
v_S	7	$\varphi_5 * (1 - q)$
v_D	8	$\varphi_6 * (1 - q)$

3) *Fehlerbehandlungsphase*: Die letzte Phase von ARA ist entworfen worden, um die Dynamik von Knotenbewegungen bzw. den Routenausfall zu behandeln. Dazu wird auf die MAC-Schicht von IEEE 802.11 zurückgegriffen. Wenn ein Knoten ein Datenpaket geschickt hat und keine Empfangsbestätigung (ACK) erhält, so wird er diesen Pfad aus seiner Tabelle löschen.

II. IMPLEMENTIERUNG MIT SENSORKNOTEN

Der *Ameisen - Routing - Algorithmus* scheint aufgrund seiner Eigenschaft des *reaktiven Routings* geeignet für MANETs aus kleineren Sensorknoten zu sein. Hinzu kommt, dass für die Anpassungen an dynamische Topologien keine ganzen Routing-Tabellen ausgetauscht werden müssen, sondern Pakete mit speziellen Flags und FANTs/BANTs im Allgemeinen ausreichen. Außerdem, unterstützt ARA die Möglichkeit, dass ein Knoten mehrere Wege zu einem Zielknoten in seiner Routing-Tabelle besitzt.

Die Implementierung von ARA wurde so vorgenommen, wie im vorherigen Kapitel beschrieben. In diesem Kapitel sollen dagegen die Besonderheiten und verwendeten Datenstrukturen dargestellt werden.

A. Verwendete Datenstrukturen

Die FANT- und BANT-Pakete basieren auf derselben Datenstruktur und enthalten die gleichen Informationen. Die übermittelten Informationen in einem FANT/BANT sind: Quellknoten (Absender), Zielknoten (Empfänger), Sequenznummer, Vorgänger-Knoten und Hopcounter.

Der Vorgänger-Knoten und der Quellknoten sind für die Erstellung eines neuen Eintrags in der Routing-Tabelle wichtig, dabei wird der Quellknoten in der Tabelle als *Zielknoten* genommen und der Vorgänger-Knoten als *Next – Hop*.

Der Quellknoten und die Sequenznummer werden genutzt, um mehrfach empfangene Pakete zu erkennen, solche Pakete werden zwar verarbeitet und erzeugen Routing-Tabellen-Einträge, werden allerdings nicht weitergeleitet. Wenn das Paket komplett ignoriert werden würde, könnten in ARA keine Multipaths entstehen. Um nun solche duplizierten Pakete zu erkennen, werden in einer *FANT – BANT – Tabelle* die Quellknoten-ID und Sequenznummer von allen empfangenen FANTs/BANTs gespeichert. Beim Erhalt eines solchen Pakets wird überprüft, ob es bereits verarbeitet wurde oder nicht. Neben dem Quellknoten und der Sequenznummer wird in der Tabelle zusätzlich ein *agingvalue* für jeden Eintrag gespeichert. Der initiale Wert dieser Variable ist 0, wird dann aber alle r Sekunden (in der Regel alle 30 Sekunden) inkrementiert. Wenn der Wert einen vorgegebenen Schwellwert ($AGE - THRESHOLD$) erreicht hat, wird der Eintrag gelöscht. Dadurch wird vermieden, dass die Tabelle vollständig gefüllt wird sowie dass nicht mehr benötigte Daten gespeichert bleiben.

Der Hop-Counter wird benutzt, den Pheromon-Wert des entsprechenden Eintrags zu initialisieren. In der Regel wird dieser Wert mit $1/Hops$ errechnet. Die Routing-Tabelle hat die Struktur, wie sie im vorherigen Kapitel beschrieben wurde. Ein Datenpaket besteht aus Quellknoten, Zielknoten, Flag, Sequenznummer und den Daten. Flag und Sequenznummer werden dazu genutzt, um zu erkennen, ob ein Paket bereits empfangen wurde und somit eine Schleife entstanden ist. Dazu werden in der *Data – Sequenz – Tabelle* die Informationen zu allen erhaltenen Paketen gespeichert - die Funktionalität entspricht dabei der *FANT – BANT – Tabelle*.

B. Funktionsweise

Wenn nun ein Knoten ein Datenpaket zu einem Zielknoten senden möchte, so durchsucht er zuerst seine Routing-Tabelle nach einem passenden Eintrag. Falls er einen passenden Eintrag gefunden

hat, wird er das Datenpaket zum nächsten Knoten (*Next – Hop*) senden. Konnte kein Eintrag in der Tabelle gefunden werden, so werden die Funktionen der Pfadentdeckungsphase genutzt. Dabei schickt der Knoten ein FANT zum Zielknoten, antwortet dieser nicht innerhalb von 30 Sekunden mit einem BANT, so wird erneut ein FANT abgeschickt. Dieser Vorgang wiederholt sich maximal fünf mal. Falls danach immer noch kein BANT vom Zielknoten empfangen wurde, so wird der Sendeversuch abgebrochen, da es keinen Weg vom Quell- zum Zielknoten gibt.

Zu Beginn der Route-Maintenance-Phase ist der Pheromonwert von allen Einträgen in der Routing-Tabelle $1/Hops$. Wenn ein Knoten ein Paket empfängt und an einen anderen Knoten weiterleitet, so wird zuerst der Pheromonwert des Eintrags zum vorherigen Knoten und Quellknoten (Link zum Absender) um ein $\Delta\varphi = 1$ erhöht und anschließend auch der Eintrag zum Zielknoten und Next-Hop (Link zum Zielknoten).

Daneben wird auch die Routing-Tabelle alle r Sekunden aktualisiert. Bei der Aktualisierung wird der Pheromonwert aller Links mit einem Faktor *pheromoneIncreasingValue* multipliziert. Wenn ein Wert dann einen bestimmten Schwellwert - $PHEROMONE - THRESHOLD$ - unterschreitet, wird er aus der Tabelle entfernt.

Beim Erhalt eines Paketes überprüft der Knoten zuerst, ob er der Zielknoten ist. Falls nicht, werden die Paket-Daten (Sequenznummer und Absender-ID) in der *Daten – Sequenz – Tabelle* gespeichert und das Paket anschließend weitergeleitet. Wenn sich herausstellt, dass ein Paket bereits empfangen wurde, d.h. dass es eine Schleife in der Route gibt, so wird dies mit dem gesetzten *FLAG – DUPLICATED* an den vorherigen Absender (Vorgänger) zurückgeschickt. Der Vorgänger löscht daraufhin den Link zu diesem Knoten, wodurch die Schleife unterbrochen wird, und sucht in seiner Routing-Tabelle nach einem anderen Weg zum Zielknoten. Falls ein weiterer Weg existiert, wird das Paket an den entsprechenden Knoten weitergeleitet, sonst wird das Datenpaket verworfen. Mit dieser Methode können Endlosschleifen vermieden werden, allerdings geht dies mit einem Paketverlust einher.

III. EVALUATION

A. Übersicht

Für die Analyse des ARA-Protokolls unter realen Bedingungen wurden eine Reihe von verschiedenen Tests durchgeführt und ausgewertet. Dabei wurden die folgenden Aspekte untersucht:

- Datendurchsatz und Fehlerrate
- Route-Discovery-Phase
- Multipath
- Schleifen
- Verhalten bei verändertem Wegwahl-Algorithmus

Die festgestellten Besonderheiten und Probleme, welche beim Experimentieren auftraten, werden in Kapitel IV näher erläutert und diskutiert.

B. Versuchsaufbau

Für die unterschiedlichen Experimente wurden bis zu 14 Sensorknoten gleichzeitig verwendet. Dabei wurde auf eine möglichst gleiche Sendeleistung bei allen Knoten geachtet, dies wurde einerseits durch das Einstellen der Sendeleistung an den einzelnen Knoten mit Hilfe des *txpwr*-Befehls erreicht, und andererseits durch das Kürzen aller Antennen auf eine einheitliche Länge. Weiterhin wurde darauf geachtet, dass der Versuchsaufbau reproduzierbar ist, durch Kennzeichnung der Sensorknoten und Markierungen der einzelnen Positionen. Dies ermöglichte, dass bei einem erneuten Test die gleiche Topologie vorhanden ist.

C. Datendurchsatz und Fehlerrate

Bei diesem Versuch sollte die Implementierung dahingehend untersucht werden, wie sie sich beim Versenden von Paketen über mehrere Hops verhält und wie hoch die Fehlerrate ist.

1) *Versuch:* Für die Durchführung der Tests, wurden unterschiedliche Topologien aufgebaut. Die Pakete wurde über eine Entfernung von ein bis drei Hops gesendet. Dabei war das Netzwerk so aufgebaut, dass ein Knoten nur mit seinen direkten Nachbarn kommunizieren konnte und es somit keine Multipaths gab, sondern nur einen möglichen

Weg. Eine 3-Hop-Topologie ist in Abbildung 4 zu sehen.

Beim Testen wurden automatisch jeweils 100

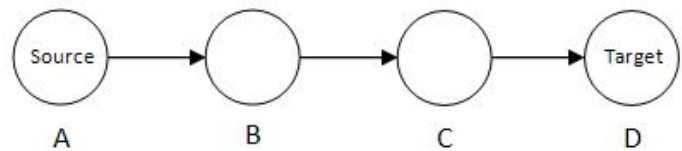


Abbildung 4. Versuchsaufbau einer 3-Hop-Topologie; Paket von A nach D

Pakete alle 300 Millisekunden mit vorgegebener Größe versendet. Im Vorfeld zu diesem Test wurde die allgemeine Sendequalität - in Bezug auf die Paketgröße - der Knoten ermittelt. Dafür wurden mehrere Pakete unterschiedlicher Größe und ohne Routing-Protokoll, d.h. direkt von Knoten A zu B, versendet. Es stellte sich heraus, dass bis zu einer Paketgröße von ungefähr 80 Byte die Übertragungsrate akzeptabel war, danach allerdings zu viele Prüfsummen-Fehler auftraten.

2) *Auswertung:* Bei der Auswertung der gemessenen Werte ist festzustellen, dass bezüglich der ARA-Implementierung kein direkter Zusammenhang zwischen der Hop-Anzahl bzw. Paketgröße und der Fehlerrate der Übertragung besteht. Es kommt zwar zu vereinzelt Verlusten von Paketen bzw. Paketen mit einer fehlerhaften Prüfsumme (CRC), aber dies kann auf die grundsätzliche Bitfehlerrate bei Funkübertragungen zurückgeführt werden, wodurch bei diesen Tests also die Implementierung sowie die Qualität der kabellosen Übertragung getestet wurde. Insgesamt lässt sich also festhalten, dass ARA auch für größere Pakete und unter Last (andauerndes Senden) geeignet ist. Dies lässt sich auch darauf zurückführen, dass für das Weiterleiten eines Paketes nur die Routing-Tabelle durchsucht werden muss um das Paket anschließend sofort weiterzusenden - ein Zwischenspeichern/Puffern für einen längeren Zeitraum ist nicht notwendig.

D. Route-Discovery

Während der Route-Discovery-Phase werden FANT- und BANT-Pakete zwischen dem Absender und Zielknoten zur Pfaderzeugung ausgetauscht.

Bei diesem Experiment sollte ermittelt werden, wie diese Phase in der Praxis verläuft und es ob zu Besonderheiten kommen kann.

1) *Versuch:* Für den Versuch wurden Netzwerke aus 10 Knoten mit verschiedenen Topologien aufgebaut und Pakete zwischen zwei Knoten, die sich jeweils am Rande befanden, gesendet (siehe Abbildung 5).

Insgesamt wurden die Tests 5-mal pro Aufbau

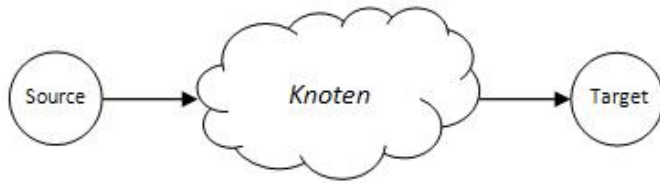


Abbildung 5. Versuchsaufbau mit mehreren vernetzten Knoten

durchgeführt, wobei es 10 verschiedenen Topologien gab. Diese haben sich in der Knotenposition sowie der Entfernung zwischen den Knoten unterschieden. Nach jedem Durchlauf wurden alle Knoten neugestartet, sodass alle Werte, wie etwa die Routing-Tabelle, sich wieder im Ursprungszustand befinden.

Dieser Versuch hatte bei allen Durchführungen nur zwei mögliche Ergebnisse: Entweder, die Route-Discovery-Phase ging auf Anhieb problemlos vonstatten, oder es konnte bei dem jeweiligen Aufbau überhaupt keine Route aufgebaut werden. Wenn die Wegfindungsphase funktionierte, dann reichte es aus, wenn der sendende Knoten ein FANT abschickte. Der Zielknoten hat daraufhin ein oder mehrere FANTs erhalten, je nachdem wieviel Knoten ihn in ihrer Sendereichweite hatten. Genauso hat es sich auch umgekehrt verhalten, der sendende Zielknoten hat ein oder mehrere BANTs vom Zielknoten empfangen. Die Zeit für solch einen Vorgang lag dabei deutlich unter einer Sekunden.

Falls die Route-Discovery-Phase keinen Pfad aufbauen konnte, dann lag es daran, dass das FANT oder BANT nicht erfolgreich durch das Netz geflutet werden konnte und es unterwegs verloren ging. Absender und Zielknoten befanden sich also nicht im gleichen Netz, sondern es existierten mehrere Netzwerke, die nicht zusammenhängend waren. Auch die Tatsache, dass ein sendender Knoten nach fünf Sekunden ohne BANT-Erhalt

erneut ein FANT versendet, konnte dies nicht überbrücken.

2) *Auswertung:* Die Ergebnisse zeigen, dass die Route-Discovery-Phase bei ARA eine einfache und effiziente Variante ist, um Pfade zwischen Sender und Empfänger aufzubauen. Außerdem werden auf diese Weise auch Multipaths ermöglicht, worauf in Abschnitt III-E noch näher eingegangen wird.

Wenn es zu Problemen kam, dann lag dies daran, dass das Netzwerk aufgrund der Sendeleistung der Knoten nicht zusammenhängend war und es somit gar keine Möglichkeit für das erfolgreiche Fluten des Netzes gab. Ansonsten erreicht man mit dem Fluten des Netzes die größte Wahrscheinlichkeit, dass Sender und Empfänger erfolgreich die FANTs und BANTs austauschen können.

E. Multipath

Beim ARA-Protokoll kann es vorkommen, dass ein Knoten mehrere Auswahlmöglichkeiten für den nächsten Knoten hat, an den ein Paket weitergeleitet werden kann. Wie bereits in Abschnitt *Implementierung* erläutert, entstehen diese Wege, indem jedes empfangene Paket der *Route-Discovery-Phase*, also FANT und BANT, einen Eintrag in der Routing-Tabelle erzeugt, und nur Pakete, die doppelt bei einem Knoten ankommen, nicht weitergeleitet werden. Diese Pakete sind an der im Paket enthaltenen Sequenznummer und der Knoten-ID des Absenders eindeutig zu erkennen. Wie es sich mit den Multipaths verhält und welche Vorteile sowie Nachteile daraus entstehen können, soll in diesem und folgenden Abschnitten näher untersucht werden.

1) *Versuch:* Um die Multipath-Eigenschaften der Implementierung zu untersuchen, wurde ein Netzwerk aus 14 Knoten aufgebaut, wobei darauf geachtet wurde, dass es möglichst zusammenhängend ist und dass Sender sowie Empfänger in einem Netzwerk sind. Anschließend wurden Pakete von einem Ende des Netzwerkes zum Anderen gesendet (siehe Abbildung 5).

Die Routen entstehen während der Route-Discovery-Phase und werden danach durch Erhöhung der Pheromonwerte verstärkt oder bei Inaktivität gelöscht. In Abbildung 7 ist ein realer

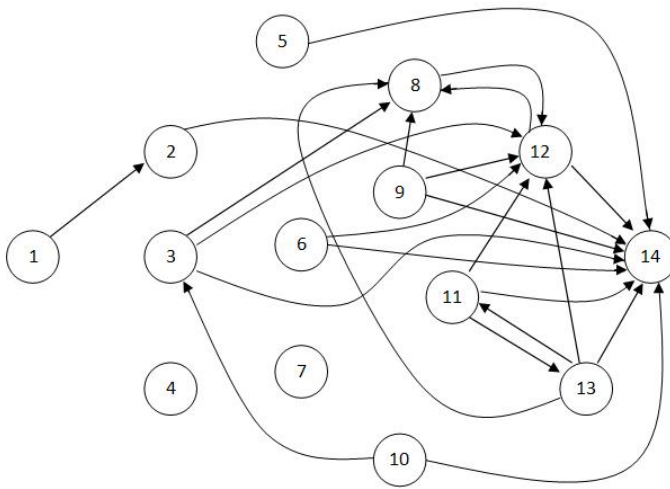


Abbildung 6. Vorhandene Routen zum Knoten 14

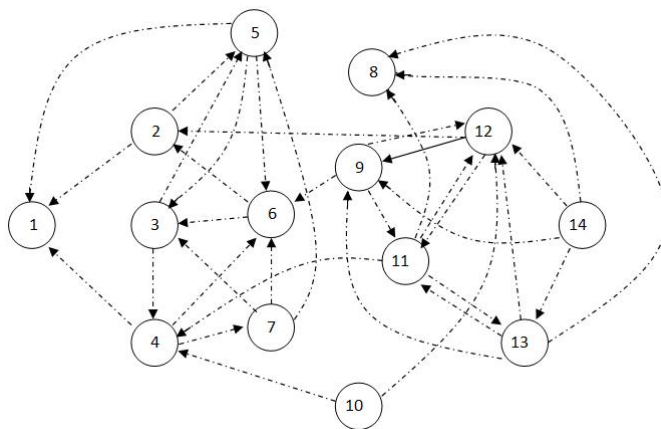


Abbildung 7. Vorhandene Routen zum Knoten 1

Teil der Topologie - bestehend aus 14 Knoten - zu sehen, diese Vernetzung ist dabei beispielhaft für alle erstellten Test-Netzwerke, da sie sich sonst nur durch einige andere Routen unterscheiden. Diese Routen sind durch das Versenden eines FANTs von Knoten 14 an Knoten 1 entstanden. Ein Pfeil von Knoten X zu Y bedeutet, dass Knoten X ein Eintrag/Weg in seiner Routing-Tabelle für Knoten 14 hat, und der nächste Hop dabei Knoten Y ist. Es ist zu erkennen, dass die meisten Knoten einen direkten Weg - einen Hop - zum FANT-Absender haben. Dies bedeutet, dass sie das FANT-Paket direkt von Knoten 14 erhalten haben. Daneben existieren allerdings auch bei fast allen Knoten alternative Wege zum Knoten 14. Diese Wege sind dadurch entstanden, dass die Knoten ein weitergeleitetes FANT erhalten haben und dahingehend ihre Routing-Tabelle erweitert haben.

Analog verhält es sich mit BANTs.

Die Anzahl der versendeten FANTs oder BANTs ist bei einem zusammenhängenden Netzwerk gleich der Anzahl der Knoten n im Netz, dies ist der Fall, da jeder Knoten nur das erste erhaltene Paket einer Route-Discovery-Phase weiterleitet, jedes weitere wird als Duplikat erkannt. Insgesamt werden also höchstens $2*n$ FANTs und BANTs versendet. Wie viel Pakete dagegen ein Knoten erhält, und wie viel Routing-Tabelle-Einträge er damit besitzt, ist abhängig von der Topologie des Netzes. Bei einem vollvermaschten Netz ist die Anzahl der Wege bzw. der empfangenen Pakete am Höchsten und liegt bei insgesamt $2*n-2$, da von jedem anderen Knoten ein FANT und BANT empfangen wird.

2) *Auswertung:* Durch das Fluten der FANTs und BANTs im Netzwerk kann schnell herausgefunden werden, über welche Knoten der FANT bzw. BANT-Absender erreichbar ist. Da ein Knoten solch ein Paket nur weiterleitet, wenn es für ihn neu ist, wird ein endloses Fluten und ein Überlasten des Netzwerkes verhindert. Die daraus entstehenden Multipaths verbessern die Robustheit von ARA gegenüber dem Auswahl von einzelnen Routen, da eventuell eine Alternativ-Route zur Auswahl steht. Zudem kann bei stark frequentierten Routen die Netzlast besser auf unterschiedliche Wege/Knoten verteilt werden, indem der Wegwahl-Algorithmus dahingehend angepasst wird, dass häufig benutzte Wege seltener ausgewählt werden oder aber auch dass alle Wege gleich oft benutzt werden, beispielsweise mit Hilfe eines *Round-Robin-Verfahrens*. Die Veränderung des Wegwahl-Algorithmus wird im Folgenden noch näher diskutiert.

Ein Nachteil bezüglich Multipaths bei ARA ist, dass Schleifen entstehen können. In Abbildung 5 ist beispielweise eine Schleife zwischen Knoten 11 und 13 zu erkennen. Durch solche Schleifen besteht die Gefahr, dass ein Datenpaket immer zwischen Knoten hin- und hergeschickt wird, ohne dass es zum Zielknoten gelangt und so wichtige Ressourcen (Energie/Batterie, Übertragungsmedium) vergeudet werden. Aus diesem Grunde wird auf das Thema Schleifen im folgenden Abschnitt III-F noch näher eingegangen.

F. Schleifen und Wegeauswahl

Schleifen im Netzwerk können bei ARA durch FANTs und BANTs entstehen - wenn ein Knoten ein FANT ein zweites Mal erhält, so wird es zwar nicht weitergeleitet, die Routing-Tabelle wird dennoch aktualisiert. Dadurch kann es passieren, dass der Empfänger-Knoten annimmt, der letzte FANT-Absender hat einen weiteren Weg zum ursprünglichen FANT-Absender, obwohl der Empfänger genau dieses Paket früher schon empfangen hat und bereits im Pfad mit aufgenommen ist. Dieses Verhalten kann auch

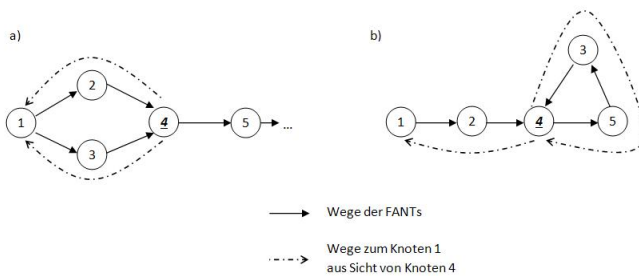


Abbildung 8. Entstehung von Schleifen

nicht durch die Sequenznummer und ID eines FANTs/BANTs verhindert werden. Denn ein Knoten kann nicht unterscheiden, ob es sich um ein Paket handelt, welches er selber bereits weitergeleitet hat oder nicht. Dieser Sachverhalt ist in Bild 8 aus Sicht von Knoten 4 dargestellt.

Es sollte nun untersucht werden, wie sich das Phänomen der Schleifen in der Praxis ausübt und wie mit ihnen umgegangen werden kann.

1) *Versuch:* Es konnte in mehreren Experimenten mit Netzwerken aus 14 Knoten gezeigt werden, dass die Auswahl des nächsten Hops aus der Routing-Tabelle - mittels eines bestimmten Wegeauswahl-Algorithmus - einen wesentlichen Einfluss auf die Bildung von Schleifen in Paket-Routen hat.

Während der Route-Discovery-Phase werden die Einträge/Pfade in den Routing-Tabellen mit einem bestimmten Wert initialisiert. Dieser Wert errechnet sich aus dem Reziproken der Hops-Anzahl, die ein FANT/BANT besitzt, also $1/\text{Hops}$. Aus diesem Grunde erhalten Routen, auf denen ein FANT/BANT mit wenig Hops (kürzerer Pfad) versendet werden einen größeren Wert, als Routen, auf denen FANTs/BANTs mehr Hops benötigen.

Wenn nun der Wegeauswahl-Algorithmus so verfährt, dass der Eintrag für den Zielknoten aus der Routing-Tabelle genommen wird, der den größten Wert hat, so entstehen keine Schleifen beim Versenden von Paketen. Dies ist der Fall, da durch solch eine Auswahl kürzere Routen immer vor Längeren bevorzugt werden - und Wege mit Schleifen haben immer einen längeren Pfad bzw. größere Hop-Anzahl. Der Knoten 4 aus Abbildung 8 b) würde demnach die direkte Route über Knoten 2 gegenüber der Route (Schleife) über Knoten 3 favorisieren.

Allerdings kann es durch Veränderung der Wegeauswahl zu einem anderen Verhalten bezüglich Schleifen kommen. Wenn der Initialwert eines Tabellen-Eintrages gleich bleibt und nur der Auswahl-Algorithmus dahingehend geändert wird, dass jede mögliche Route zum Zielknoten gleich oft verwendet wird, so können Schleifen auf Paket-Routen entstehen. Denn nun ist die Auswahl der Route nicht mehr vom Initialwert - also der Pfadlänge - abhängig, sondern alle vorhandenen Routen werden gleich behandelt. Es kann allerdings auch beobachtet werden, dass es nicht zu Endlosschleifen auf den Routen kommt. Denn jeder Knoten wählt jedes Mal einen anderen Weg, wodurch es zwar zu Schleifen kommen kann, das Paket aber irgendwann beim Empfänger ankommt. Aus dem Beispiel in Abbildung 8 b) könnte eine mögliche Paket-Route von Knoten 4 zu Knoten 1 folgendermaßen aussehen: 4 - 3 - 5 - 4 - 2 - 1. Es gibt zwar eine Schleife, aber das Paket erreicht sein Ziel.

2) *Auswertung:* Die einfachste und effizienteste Methode Schleifen bei ARA zu vermeiden, ist die Verwendung eines Wegeauswahl-Algorithmus, der die kürzesten Routen bevorzugt - Schleifen und lange Routen werden dadurch automatisch nicht benutzt. Allerdings entsteht dadurch die Gefahr, dass bestimmte Routen sehr stark benutzt werden und es zu Überlastungen auf diesen Strecken kommen kann. Außerdem geht dadurch etwas von der Flexibilität des Protokolls verloren, denn es werden nicht mehr andere Pfade genutzt, welche auch zu Verfügung stehen würden. Nur wenn es zum Ausfall der Route kommt, könnte ein anderer Weg gewählt werden. Wobei bei diesem neu gewählten Weg wiederum die Gefahr einer Schleife besteht. ARA bietet zwar eine einfache

Schleifenbehandlung mittels *FLAG-DUPLICATED* an, diese ist aber meist nicht ausreichend. Denn wenn ein Knoten ein Paket ein zweites Mal erhält, dann kann er dem Vorgänger das Paket mit diesem gesetzten FLAG zurücksenden, woraufhin dieser den Link zum Knoten löscht und einen neuen Weg/Knoten heraussucht. Wenn allerdings dieser neue Knoten das Paket auch zurückschicken sollte, oder allgemein kein Link mehr vorhanden ist, so kann der Knoten nichts mehr mit Paket anfangen, da er nicht mehr weiß, von wem er das Paket empfangen hat. In diesem Fall bliebe nur noch die Möglichkeit des Flutens.

Bei einem anderen Wegeauswahl-Algorithmus, können zwar die vorhandenen Multipaths besser genutzt werden, aber Schleifenbildungen werden wahrscheinlicher, wodurch eine umfassendere Schleifenbehandlung notwendig wird. Allerdings haben Algorithmen wie das Round-Robin-Verfahren wesentliche Vorteile bei *unidirektionalen Links* (siehe Abschnitt IV-A), da damit die Auswahlquote bei der Paketzustellung in einigen Fällen verringert werden kann. Dies wird im folgenden Kapitel noch weiter erläutert.

IV. BESONDERHEITEN

Während des Testens und der Analyse des ARA-Protokolls haben sich einiger Besonderheiten diesbezüglich herausgestellt, diese sollen in diesem Kapitel untersucht werden.

A. Unidirektionale Links

Beim Testen bzw. beim Aufbau einer gewünschten Topologie stellten sich häufig unidirektionale Links und unterschiedliche Sendeleistung als besondere Bedingungen heraus. Unidirektionale Links bedeuten, dass die Sendeleistung von zwei benachbarten Knoten nicht gleich ist und somit ein Knoten A Pakete vom Anderen (B) empfangen kann, während der Knoten B keine Pakete von A erhält. Das Beispiel ist in Abbildung 9 dargestellt. Dies ist eine Eigenschaft, die nicht ungewöhnlich bei Sensornetzwerken bzw. bei Funkübertragungen im Allgemeinen ist. Dennoch hat dies einen erheblichen Einfluss auf die ARA-Implementierung. Denn bei ARA werden die Routen *rückwärts* aufgebaut, d.h.

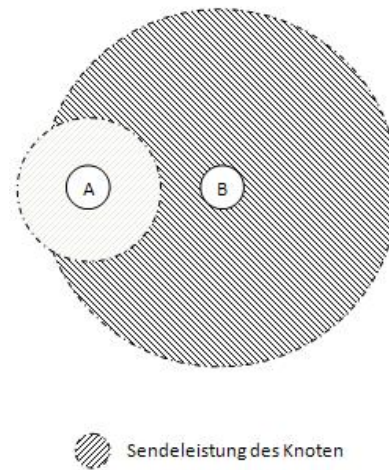


Abbildung 9. Unidirektionale Links

wenn ein Knoten ein FANT/BANT von einem Knoten empfängt, dann weißer, dass er über diesen Knoten den Absender erreichen kann. Ein Problem entsteht nun, wenn ein unidirektionaler Link wie in Abbildung 9 existiert und Knoten B ein FANT oder BANT von X an Knoten A sendet. Knoten A empfängt das Paket und wird nun alle Pakete, die er an X senden will, an B weiterleiten. Allerdings kann B die Pakete von A nicht empfangen und die Daten gehen verloren. Hinzu kommt noch, dass bei den Sensorknoten kein MAC Acknowledgement nutzbar war (siehe dazu IV-B). So konnte es vorkommen, dass bei einer aufgebauten Route alle gesendeten Pakete verloren gingen, weil es einen unidirektionalen Link auf dem Weg gab und der Absender nichts davon erfährt. Wenn dagegen die volle Funktionalität der MAC-Schicht gegeben wäre, so würde eine fehlgeschlagene Übertragung durch ein ausbleibendes ACK bemerkt werden, und der Link könnte gelöscht und das Paket an einen anderen Knoten weitergeleitet werden.

In diesem Zusammenhang zeigt sich noch der Vorteil von einem gleichverteilten (Round-Robin) Wegeauswahl-Algorithmus. Bei einer Auswahl des Weges anhand des höchsten Pheromonwertes, ist es wahrscheinlicher, dass ein unidirektionaler Link genommen wird. Denn wenn es im Netzwerk einen Knoten B mit starker Sendeleistung gibt, der viele Knoten erreichen kann, so werden diese Knoten Pakete an Knoten B weiterleiten wollen, da dieser mit wenig Hops erreichbar erscheint. Bei einem anderen Wegeauswahlalgorithmus kann das Verhalten so abgeändert werden, dass nicht alle

Pakete über die gleichen Links versendet werden, sondern Verschiedene benutzt werden. Dadurch steigt die Wahrscheinlichkeit, dass nicht sämtliche Pakete verloren gehen, sondern wenigstens einige ihr Ziel erreichen.

Eine Lösung für dieses Problem wäre zum einen die Funktionalität der MAC-Schicht zu nutzen, sofern sie zur Verfügung steht, oder selbständig Sendebestätigung zu Versenden, und bei ausbleibenden Bestätigungen den Link zu deaktivieren. Eine weitere Möglichkeit wäre es, die Links gleich während der Route-Discovery-Phase zu bestätigen. Falls ein Knoten ein FANT/BANT erhält, so schickt er zum Absender ein Paket, auf welches dieser Antworten muss. Erst wenn diese Bestätigung eintrifft, wird der Link tatsächlich aktiviert, ansonsten wird er verworfen.

B. MAC-Bestätigungen

Wie bereits erwähnt, konnten bei den Sensorknoten nicht alle Funktionen der MAC-Schicht benutzt werden, dazu gehört auch die Empfangsbestätigung von versendeten Paketen (Acknowledgements). Diese Bestätigungen sind allerdings wichtig für die Route-Maintenance, also die Verwaltung der Links in den Routing-Tabellen. Ohne solche ACKs können fehlerhafte Links aber nur schwer erkannt und gelöscht werden.

V. FAZIT

Mit Hilfe von ARA ist es möglich, eine Kommunikation zwischen mehreren Sensorknoten in MANETs aufzubauen. Die Implementierung erfordert nur wenig Verwaltungsoverhead bei den Knoten und ist effizient gestaltet - es existiert pro Knoten nur eine Routing-Tabelle, sowie einige Datenstrukturen um die FANTs/BANTs zu verwalten. Auch die Link-Pflege ist energiesparend gestaltet, da nur die eigenen Routing-Tabelle-Einträge periodisch inkrementiert werden und keine extra Pakete dafür versendet werden müssen. Allerdings ist das Fluten des Netzwerkes mit FANTs/BANTs eine große Belastung für das Netz, gerade für große und mobile Netzwerke.

Wenn die darunterliegende MAC-Schicht vollständig genutzt werden kann, so bietet ARA auch eine recht hohe Zuverlässigkeit bezüglich der Datenübertrag. Hinzu kommt auch, dass

ARA sehr flexibel bei dynamischen Netzen und verändernden Topologien ist, da aufgrund der Multipath-Unterstützung ein alternativer Weg genutzt und bei fehlenden Routen auch die Route-Discovery-Phase schnell durchgeführt werden kann. Allerdings besteht die Gefahr, dass sich Schleifen bilden können. Wobei eine umfassende Schleifen-Behandlung die Komplexität der Implementierung erhöhen würde.

LITERATUR

- [1] Mesut Günes, Martin Kähler, Imed Bouazizi Ant-Routing-Algorithm (ARA) for Mobile-Multi-Hop ad-hoc networks new Features and results Proceedings of the Med-Hoc Net 2003 Workshop Mahdia, Tunisia 25-27 June, 2003
- [2] Charles E. Perkins IBM, T.J. Watson Research Center Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers Hawthorne, NY 10562
- [3] RFC 3626, T. Clausen, P. Jacquet, Optimized Link State Routing Protocol (OLSR) October 2003
- [4] RFC 4728 D. Johnson, Y. Hu, D. Maltz, The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4 Microsoft Research February 2007
- [5] Zygmunt J. Haas, Marc R. Pearlman, Prince Samar, The Zone Routing Protocol (ZRP) for Ad Hoc Networks Cornell University January 2003