VRR - Virtual Ring Routing

Krzysztof Ibek
Seminar „Internet Routing“ ,
Technische Universität Berlin

WS 2007/2008 (Version vom 24. Januar 2008)

1 Einleitung

Dieses Dokument beschreibt ein Verfahren zum Routing in ad hoc Wireless Netzwerken: Virtual Ring Routing VRR. VRR zeichnet es aus, dass es Routen bestimmen kann, ohne dabei das Netzwerk zu fluten. Hilfreich dabei sind eingesetzte Techniken aus dem Bereich von Distributed Hash Tables DHT. VRR stellt eine sowohl adressenbezogene, als auch Punt-zu-Punkt-Kommunikation sicher. Es wird direkt über dem Linklayer im Protokollstack implementiert.

1.1 Motivation

Der Schlüssel für ein erfolgreiches Routing durch ein Netzwerk ist die Kenntnis seiner Topologie. Die wichtigsten bisher eingesetzten Routing-Protokolle im Wireless-Bereich haben einige Nachteile. Um die Topologie des Netzes kennenzulernen, überfluten sie es mit Erkundungsnachrichten. Dieses Verhalten verursacht einerseits Übertragungsverzögerungen, anderseits stellen diese vielen Nachrichten einen nicht zu vernachlässigenden Protokoll-Overhaed dar. Knotenausfälle oder erhöhte Mobilität der Knoten verstärkten noch diese beiden Nachteile. Aufgrund der internen Adaptationen funktionieren einige Protokolle sehr gut bei statischen Szenarios, andere versuchen, die Mobilität der Knoten zu meistern. Auf der anderen Seite kann diese Art der Spezialisierung zu Performanceeinbußen in den nicht angepassten Bereich führen, wie es später zu sehen sein wird. Hier setzt VRR an, indem es das Fluten umgeht und im Vergleich zu anderen Protokollen eine gute Performance zeigt, sogar über verschiedene Szenarien hinweg.

1.2 Aufbau des Dokuments

Die Kenntnis der zugrundeliegenden Mechanismen bei der Kommunikation in Wireless-Netzwerken hilft, die zum Teil abweichenden Designentscheidungen des VRR nachzuvollziehen. Die Grundlagen gehen, neben den Mechanismen, auch auf die Begriffsklärung und Klassifizierung der Netzwerke in kabelloser Umgebung ein. Danach stellt das Papier VRR vor. Neben der Interna wird auch die Performance des VRR im Vergleich zu anderen Protokollen aus diesem Bereich untersucht.

2 Grundlagen

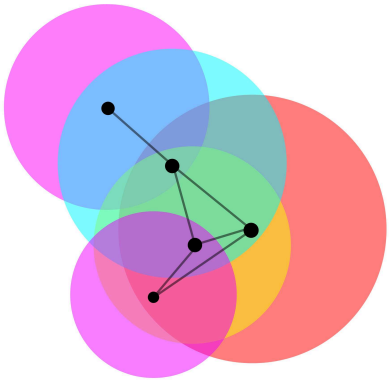

Abbildung 1: Maschennetzwerk. Knoten mit Sendebereichen und Kommunikationspfaden

Es gibt grundsätzlich zwei Möglichkeiten in kabelloser Umgebung eine Kommunikation zwischen den lose angeordneten Geräten herzustellen. Eine Möglichkeit ist, einen zentralen Zugangspunkt in der Umgebung der Geräte zu schaffen, der eine Rolle der Vermittlungsstelle übernimmt, einen Accesspoint. Dieses Verfahren erfordert, dass die beteiligten Knoten sich zuerst bei dem Accesspoint registrieren. Dabei erhalten sie eindeutige Adressen, wodurch die Nachrichten den jeweiligen Sendern und Empfängern durch den Accesspoint zugeordnet werden können. Diese Art der Kommunikation setzt voraus, dass alle kommunizierenden Geräte sich in dem Sendebereich des Accesspoints aufhalten. Der zweite Ansatz, der hier in Hinsicht auf VRR weiter vertieft wird, ist ein dezentraler. Dabei wird den Knoten als gleichwertige Partner selbst überlassen, mithilfe eines geeigneten Protokolls die Kommunikation organisieren zu lassen. Diese Ad-hoc-Verbindungen können zwischen zwei Knoten (P2P) stattfinden. Ein Maschennetzwerk entsteht, wenn mehrere Knoten miteinander verbunden sind und die Kommunikation zwischen Knoten stattfindet, die nicht in der gegenseitigen Reichweite sind. Hier ist VRR angesiedelt.

2.1 Kommunikation im Maschennetzwerk

Die Nachricht wird so lange vom Nachbar zum Nachbar weitergereicht, bis der Empfänger diese bekommen hat. Es muss lediglich eine Route zu dem Empfänger vorhanden sein, damit die Nachricht ankommt. Die Abbildung 1 zeigt eine mögliche Anordnung der Konten, und wie eine Vermaschung untereinander aussehen kann. Bei dem dezentralen Ansatz gibt es demnach keinerlei Instanz, die alle Informationen über das zustande gekommene Netzwerk innehat. Initial kann der Sender Informationen nur mit seinen Nachbarn austauschen. Möchte er jetzt an einen anderen Knoten eine Nachricht schicken, so steht er vorerst vor zwei Problemen: Wie kommt er an die Adresse seines Empfängers und, sollte er diese erlangen, so muss auch seine Nachricht irgendwie durch das Maschennetz transportiert werden. Die Adressen müssen im gesamten Netzwerk eindeutig sein. Sie können fest zugewiesen werden, oder die Vergabemechanismen können sich auch aus dem jeweiligen Übertragungsprotokoll ergeben. Die Route ist dem Sender vorerst typischerweise nicht ersichtlich. Erschwe-

rend dabei ist, dass die Knoten entlang der Route sich bewegen, oder gar ganz ausfallen können, wodurch die Route selbst dynamischen Veränderungen unterliegen kann.

2.2 Klassifizierung der Routingprotokolle

Abhängig von der Lösung des Adressierungs- und Routenfindungsproblems gibt es für jedes Problem jeweils zwei Möglichkeiten ein Protokoll zu klassifizieren. In Hinsicht auf Adressierung gibt es positionsbasierte, als auch topologiebasierte Unterscheidung[**?**]. Positionsbasierte Protokolle verwenden als Adressen Geo-Positionen (z. B.: GPS-) Positionen. Der Sender kann anhand der Koordinaten die Richtung seines Empfängers bestimmen, somit auch den nächsten Nachbar, der an der Route beteiligt ist. Dieser verfährt analog, bis die Nachricht beim Empfänger ankommt. Für statische Szenarien wird das gut funktionieren, da die Adresse des Empfängers gleich bleibt. Sollte Mobilität der Knoten berücksichtigt werden, müssen noch logische Adressen an die Knoten vergeben werden, um Adressierbarkeit sicherzustellen. Diese Maßnahme erhöht aber die Komplexität der Routings. Bei der topologiebasierenden Klassifizierung werden die Positionen im Netz durch dessen Erkundung herausgefunden. Die einzelnen Knoten sammeln Informationen über die Nachbarschaft, indem sie selbst HELLO-Nachrichten versenden und empfangen. So enthalten sie Informationen über die Nachbarschaft des Senders. Die Gesamtheit der Nachrichten ergibt die Topologie des Netzes und offenbart die alle Routen zu den jeweiligen Knoten. Diese globale Sicht ist in einem Maschennetzwerk für einen einzelnen Knoten nicht verfügbar. Möchte er eine Nachricht an andere Knoten senden, muss er diese bestehenden Routen zuerst erkunden. Wann die Routenerkundung stattfindet, ergibt die nächste Unterscheidung. Proaktive Protokolle erkunden noch vor der eigentlichen Datenübertragung alle Routen von jedem Knoten zu jedem potenziellen Empfänger. Es hat den Vorteil, dass bei einem Sendeversuch es keine Verzögerungen gibt, da der Pfad zum Empfänger bereits bekannt ist. Das Herausfinden und die unumgänglichen Aktualisierungen der Pfade verursachen viel Verkehr im Netzwerk, viele von den herausgefundenen Pfaden werden bei den späteren Übertragungen nicht benutzt, da nicht jeder Knoten mit jedem kommuniziert. In statischen Netzen wird es gut funktionieren, da die Routen nur einmal berechnet werden müssen, und dann nur noch selten. Bei vielen Änderungen in der Topologie werden die zur Routenberechnung benötigten Kontrollpakete auf Kosten der Nutzübertragung das Netzwerk fluten. Reaktive Protokolle erkunden die gerade benötigten Routen erst dann, wenn ein Knoten eine Nachricht senden möchte. Dadurch werden im Gegensatz zu dem proaktiven Verfahren viele Nachrichten eingespart. Erkauft wird es mit einer Verzögerung, die vor dem Senden entsteht. Es ist die Zeit, die nötig ist, um die Route zum Empfänger herauszufinden. Doch es gibt auch hier Grenzen. Bei einer eher statischen Topologie kann mit der Zeit ein proaktives Verfahren günstiger sein, da die Routen nur einmal berechnet werden. Daneben gibt es noch Mischformen, die die jeweiligen Nachteile umgehen möchten. Hierarchisch-Hybride-Verfahren bilden z. B. Gruppen von Knoten. Die Gruppen selbst sind mobil, wodurch zur Kommunikation zwischen den Gruppen das reaktive Verfahren eingesetzt wird. Zwischen den Gruppenmitgliedern wird das proaktive Verfahren eingesetzt, da sie untereinander eher statisch vernetzt sind. Die Gruppenmitglieder können untereinander ohne durch das reaktive Verfahren verursachte Verzögerungen kommunizieren, und das Herausfinden aller Routen innerhalb der Gruppe ist nicht so aufwendig wie im gesamten Netzwerk. Der Pfad zu den potenziell seltener kontaktierten anderen Gruppen wird reaktiv ermittelt, wodurch die

durch dieses Verfahren verursachte Verzögerung seltener ins Gewicht fällt. Das funktioniert so weit gut, solange die Knoten sich an die vorgegebenen Szenarien halten. Die Gruppen können hierarchisch angeordnet sein.

2.3 Pfadwahl

Bevor die Route vom Sender zum Empfänger durch ein Protokoll bestimmt wird, gibt es verschiedene Ansätze die Knoten zu einer Route zusammenzufügen. Typischerweise gibt es nicht nur einen Weg, der vom Sender zum Empfänger entlang der benachbarten Knoten führt. Wie in der Abbildung 1 zu sehen ist, kann der rechte Knoten übersprungen werden, da die Sendebereiche benachbarter Knoten sich überschneiden. Je nach dem, welche Strategie gewählt wird, kann die Verbindung qualitativ variieren, sich somit positiv als auch negativ auf den übertragenen Datendurchsatz auswirken. In dem Zusammenhang werden im Folgenden einige Mechanismen angesprochen[**?**]: HOP Count (HOP) setzt darauf, dass eine Route mit der niedrigsten Anzahl an beteiligten Knoten, die Pakete am schnellsten durch das Netz transportiert. Bei steigender Entfernung sinkt jedoch die Verbindungsqualität, was bei diesem Ansatz zu Einbrüchen in dem Durchsatz führen kann. Anderer Ansatz schaut sich die Round Trip Time (RTT) zwischen den Nachbarknoten an. Dabei wird sowohl die Qualität der Leitung, aus auch die Auslastung des Nachbarknotens festgestellt. Eine Route, die auf den kleinsten RTTs basiert, könnte aufgrund der variierenden RTTs instabil sein. Die Messung von RTT berücksichtigt nicht die Bandbreite der Leitung. Expected Transmission Count (ETX)[**?**] misst die Qualität der Leitung in der Nachbarschaft, indem festgestellt wird, wie viele Pakete bei der Kommunikation mit dem Nachbarknoten durchgekommen sind. RTT und ETX benötigten Kontrollpakete, die einerseits den Overhead erhöhen, anderseits zur Überlastung des Netzwerks beitragen. HOP wird nur in sehr engmaschigen Netzen gut funktionieren.

2.4 Einsiedlung im Protokollstack

Letztendlich sprechen mehrere Punkte dafür, ein Routingprotokoll wie VRR als 2.1 Schicht im Protokollstack zu implementieren, aber warum? Maschennetzwerke sind eine Variante der Peer2Peer-Netzwerke. Aus diesem Bereich ist Distributed Hash Table DHT bekannt. Diese verteilte Datenstruktur weist hier benötigte Merkmale auf: eine ID, die die Funktion der eindeutigen Adresse übernehmen könnte. Außerdem beinhaltet sie auch Informationen über die Nachbarknoten, womit ein Overlaynetzwerk auf der Anwendungsschicht aufgespannt wird. Leider setzt sie eine funktionierende Transportschicht voraus, die ohne Routing nicht funktioniert. Somit muss man die Ansiedlung des zu implementierenden Protokolls tiefer im OSI-Schichtenmodell ansetzen. Möchte man auf bereits bewährte Techniken wie TCP nicht verzichten, geht man tiefer, bis kurz über dem Linklayer. Einerseits kann man von hier aus nach unten auf bereits verbreiterte Übertragungstechniken wie 802.11 zurückgreifen könnte, anderseits können die höheren Schichten unangetastet bleiben.

3 Virtual Ring Routing

Nachdem die grundsätzlichen Funktionsweisen eines Routingprotokolls in Maschennetzwerken beschrieben wurden, lassen sich die in VRR abweichend getroffenen Designentscheidungen besser einordnen[**?**].

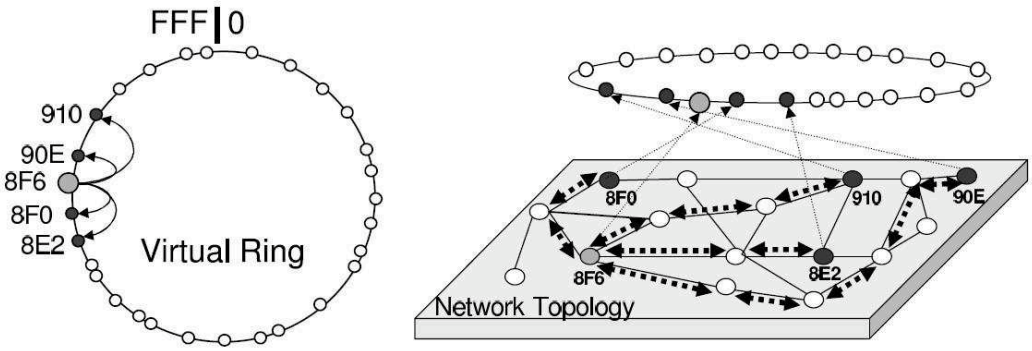

Abbildung 2: VRR links Ringaufbau, rechts Bezug zur physikalischen Topologie

3.1 Überblick

VRR ist in erster Linie ein Protokoll zum Sicherstellen des Routings in Ad-hoc-Netzwerken, ist aber auf dieses Einsatzfeld nicht beschränkt. Durch das abstrakte Design sind viele Einsatzfelder möglich. Auch die von VRR geforderte Adressierung ist mir vielen Konzepten kompatibel, sodass durch eine mögliche Eingliederung der VRR in den Stack der bestehenden Übertragungsprotokolle, weitestgehend Transparenz für die aufrufenden Schichten bestehen bleibt. Es handelt sich um logische Adressen, sodass sie Adressierung von dem Umfeld (Hardware, Topologie) nicht eingeschränkt werden kann. Es ist denkbar, das VRR IP-Adressen als IDs benutzt, sodass TCP über dem VRR funktionieren kann. Dadurch wäre ein Einsatz des VRR auch in Internet möglich. VRR ist in Protokoll, das die Ideen aus dem bereits vorgestellten Techniken übernimmt, vermeidet dabei die dadurch entstehen Nachteile wie z. B.: Fluten des Netzwerkes. Einen sehr großen Einfluss auf VRR hatte auch DHT. DHT-Konzepte findet man bei der Adressierung und der Art der Netzbildung im VRR. Über DHT hinaus ermöglicht VRR Punkt zu Punkt Verbindungen. Die einzelnen Knoten speichern Informationen zu Ihrer eigener Sicht auf das Netzwerk, genauer auf ihre Nachbarschaft, außerdem nur den für sie relevanten Anteil der Route. In Anlehnung an DHT sind es einerseits virtuelle Daten, anderseits, um das Routing zu realisieren, auch Informationen über die physikalischen Nachbarknoten. VRR unterstützt weder Broadcast noch Multicast. Bei der Adressierung setzt VRR auf logische virtuelle IDs. Es sind positive ganze Zahlen, die eindeutig sein müssen. Sie sind unabhängig von der topologischen Lage im realen physikalischen Netzwerk. Alle Knoten werden zu einem virtuellen Ring angeordnet, was dem

Protokoll den Namen gegeben hat. Nachdem alle Knoten ihrer Adresse entsprechend, in einer Reihe geordnet wurden, schließen die Knoten mit der kleinsten und der größten Adresse den Ring, indem sie zu virtuellen Nachbarn werden. Da die Adressen zufällig vergeben werden, wird die virtuelle Anordnung mit der physikalischen Topologie typischerweise wenig gemeinsam haben. Die logischen Nachbarn können sich physikalisch in der Regel nur über Zwischenknoten erreichen. Abbildung 2 verdeutlicht die beiden, gerade beschriebenen Beziehungen. Das Routing geschieht auf der virtuellen Ebene, die Pakete werden von einem virtuellen Nachbar zum anderen übermittelt. Um die physikalische Umgebung zu erkunden, sendet jeder Knoten periodisch eine HELLO-Nachricht. Hier übernimmt VRR ein Merkmal der früher beschriebenen topologiebasierenden Verfahren. Ein HELLO beinhaltet den Status des Knotens mit Informationen über physikalische Nachbarschaft und der eigenen Adresse. Durch das Hören der HELLOs anderer Knoten bekommt der Knoten einen Eindruck über seine physikalische Umgebung. Eigene HELLOs machen ihn für die Nachbarn sichtbar. Die Abstände sind von der Implementierung abhängig, oft wird 1s festgelegt.

3.2 Datenstrukturen

Jeder der Knoten unterhält, neben der Forwardingtabelle, zwei wichtige Datenstrukturen, die für das Routing entscheidend sind. Sie spiegeln jeweils die virtuelle und die physikalische Sicht des Knotens auf seine Nachbarschaft wieder. In der Datenstruktur pset sind die physikalischen Nachbarn des Knoten gespeichert. Sie werden durch den Austausch von HELLOs gesammelt. Sollten jetzt alle gefundenen Knoten als Nachbarn betrachtet werden? Wie in der Abbildung 1 zu sehen ist, hat jeder Knoten eine bestimmte Reichweite. Bei konstanter Sendeleistung hängt die Anzahl der physikalisch erreichbaren Knoten von deren Verteilungsdichte ab. Um mehr Knoten erreichen zu können, muss die Sendeleistung erhöht werden. Bei einer Übertragung würden viele alternative Knoten, somit potenzielle Übertragungswege zur Robustheit beitragen. Anderseits würden Verbindungen zu weit entfernten Knoten einen schlechteren Durchsatz haben, was sich auf die Gesamtperformance des VRR negativ durchschlagen könnte. Dieser und andere ähnliche Aspekte wurden bereits in dem Abschnitt über Grundlagen diskutiert. Dort wurden auch Mechanismen zur Lösung vorgestellt. Tatsächlich sieht VRR diese Auswahlmechanismen vor, überlässt es die Wahl der Implementierung, und lässt es an dieser Stelle offen. Dennoch, die Leitungsqualität, als Summe dieser Überlegungen, spiegelt sich in einer festzulegenden Schranke nieder. Alle Knoten, deren Leitungsqualität diese Schranke übertreffen, werden zu dem pset hinzugefügt. Die Leitungsqualität wird in die später beschriebene Forwarding-Entscheidung einbezogen. Angelehnt an DHT unterhält jeder Knoten Informationen über seine virtuellen Nachbarn namens vset. Dabei wird die Anzahl der zu merkenden Nachbarn über von der Implementierung zu definierende Kardinalität r bestimmt. $r/2$ Informationen entfallen je für die Vorgänger- und Nachfolgerknoten. Je mehr virtueller Nachbarn in dem vset gespeichert sind, desto größer ist die Ausfallsicherheit des Netzes: Mehr ausgefallene Knoten können überbrückt werden. Auch die Komplexität wird dadurch erhöht. Für die Abbildung 2 ist r=4.

3.3 Forwarding

Routingentscheidung geschieht bei VRR lokal in den einzelnen Knoten anhand der virtuellen ID des Zielknoten. Falls die eingetroffene Nachricht nicht für den Knoten selbst bestimmt

ist, leitet er sie weiter an denjenigen Knoten aus seinem vset, dessen ID am dichtesten an der ID des Zielknoten liegt. Dabei kann VRR sich auch einiger Optimierungen bedienen. Die Forwarding-Tabelle spiegelt das Verhalten wieder.

Start	End	next>Start	next>End	Pfad-ID	
vk-2	vk	npk-2	NULL	01	Eintrag vom virtuellen Vorgänger zum Knoten. Es gibt keinen nächsten
vk-1	vk	npk-1	NULL	01	vk ist der Schlusspunkt, daher wird NULL für Nachfolger eingetragen.
vk	vk+1	NULL	npk+1	01	Analog Einträge zu den virtuellen Nachfolgen.
vk	vk+2	NULL	npk+2	02	ID wird vom Startknoten bestimmt, dieser muss die ID's unterscheiden.
vk	npk+1	NULL	npk+1	FF	Optimierung: ein Hop-Einträge, Sonder-ID FF bei Optimierungen
vk	...	NULL	...	FF	Weitere Einträge für 1- und 2 Hops
...	...	...	...	...	Einträge für Routen die durch vk führen

Tabelle 1: Routigtabelle für Knoten vk bei r=4. Abkürzungen : v-virtueller Nachbar k-Knoten n-nächster p:physikalischer Nachbar +/- n: offset zu vk in dem virtuellen Ring

In der Routing Tabelle sind Paare von Endpunkten als eine Verbindung eingetragen. Ein Eintrag besteht aus der Adresse des Startknotens, Endknotens, als nächstes gibt es Einträge von physikalischen Nachbarn auf dem virtuellen Weg zum Start- bzw. Endknoten. Dabei sind die Start- und Endknoten die virtuellen Nachbarn in r. Die Vorgänger (vk-x) nehmen jeweils die Position des Startknotens ein, der Ich-Knoten (vk) ist dann der Endknoten. Für die Nachfolger (vk+x) ist analog der vk der Startknoten. Jeder Eintrag wird durch eine Pfad-ID ergänzt, die von Startknoten vergeben wird, somit ist die Pfadunterscheidung nur zusammen mit dem Startknoten eindeutig. Die Tabelle 1 vermittelt einen Eindruck über den Aufbau der Routingtabelle. Außer den dort ersichtlichen Einträgen speichert VRR noch die Pfade, die durch vk durchgehen. Durch das Abhören der HELLOs bekommt der Knoten auch Informationen über die Identitäten der Nachbarsnachbarn. An dieser Stelle ist es VRR möglich, Optimierungen der Pfadlängen durchzuführen. Wie in der Tabelle 1 zu sehen ist, werden die 1-Hop-Entfernungen gespeichert. Durch das Durchsuchen der psets der Nachbarn ist es möglich, hier auch 2-Hop-Pfade einzufügen. Die in der Routingtabelle durch VRR eingetragenen Routen sind symmetrisch, in beide Richtungen begehbar. Wenn ein Knoten eine Nachricht zum Weiterleiten bekommt, schaut er, ob er selbst der Empfänger der Nachricht ist, wenn ja, liefert er diesen an höhere Schicht ab. Andernfalls schaut er sich die Zieladresse, also die virtuelle ID des Empfängers, an. Sie wird mit den Spalten End bzw. Start (Symmetrie der Route) in der Routingtabelle verglichen. Es kann jetzt passieren, dass es dort keinen Treffer gibt, oder mindestens einen. Falls es keinen Treffer gibt, ist der Empfänger weder der virtuelle Nachbar, noch ist er in den Optimierungseinträgen (1 und 2-Hops) zu finden. Dieser übermittelnde Knoten befindet sich dann mitten drin in der virtuellen Route. Es gilt jetzt, einen geeigneten virtuellen Knoten für den nächsten Hop auszuwählen. VRR sucht jetzt in dem vset nach einen Knoten, dessen ID den geringsten Abstand zu der ID des Endknotens hat: IDmin. Virtuell wird die Nachricht an diesen Knoten geleitet. Die Eigenschaft, dass alle Knoten miteinander virtuell geordnet verbunden sind, stellt sicher, dass die Empfänger-ID nach endlicher Schrittanzahl gefunden wird. Physikalisch wird die Nachricht an den Eintrag next¿Start bzw. next¿End geschickt, je nach dem welcher Eintrag für IDmin übereinstimmt. Ein Beispiel bezogen auf Abbildung 2: Bekommt der Knoten 90E eine Nachricht für 8E1, so leitet er sie virtuell zuerst an 8F0 weiter. 8F0 hat 8E1 bereits in eigenem vset. Anderer Fall ist, dass der übermittelnde Knoten den Empfänger bei sich in der Routingtabelle findet.

Bei mehreren Treffen wird der Pfad mit der höchsten ID bevorzugt. Die höchsten Pfad-IDs haben Einträge, die auf die Optimierungen (1 und 2-Hops) zurückgehen. Auf diese Weise verkürzt VRR die Anzahl der Hops einer Route. Dieses Verfahren ist nicht mit dem in der Einführung vorgestellten Hop Count (HOP) vergleichbar, das die Routenauswahl anhand der niedrigsten Anzahl der Hops forciert. Wir erinnern uns, in den pset werden nur diejenigen physikalischen Nachbarn eingetragen, deren Verbindung eine bestimmte Qualität erreicht hat, also physikalisch nicht sehr weit entfernt sind. Der Nachteil vom HOP schlägt hier nicht durch!

3.4 Zugangsverfahren

Die Topologieänderung eines Netzes bei einem Knotenzugang meistert VRR ohne viele Routen ändern zu müssen, auch ohne Fluten. Möchte ein Knoten x dem Netzwerk beitreten, tauscht er zuerst die HELLOs mit den sich bereits im Netzwerk befindenden Knoten aus. Hört er keine Nachbarn, bildet er sein eigenes Netzwerk. Falls er HELLOs von aus der Umgebung empfängt, sucht er sich einen Knoten heraus und benutzt ihn nachfolgend, um dem Netzwerk beizutreten. Dieser ausgewählte Knoten p fungiert für ihn als Proxy. x sendet eine Setup-Request-Nachricht an den Proxy p, mit der eigenen virtuellen ID als Empfänger. Diese Nachricht wird, wie wir es kürzlich gesehen haben, an den Knoten y weitergeleitet, dessen ID am dichtesten von x entfernt ist. Die Setup-Request-Nachricht landet also bei dem zukünftigen virtuellen Nachbar von x. Y fügt x zu seinem vset hinzu und antwortet mit einer Setup-Nachricht, die über p an x geleitet wird. Alle Knoten entlang der virtuellen Route zwischen y und x fügen den neuen Pfad mit der entsprechenden Pfad-ID der eigenen Routingtabelle hinzu. Es ist eine Route entstanden, ohne das Netzwerk zu fluten. Nach dem Erhalt von der Setup-Nachricht von y, fügt x y zu seinem vset hinzu. Die gerade angekommene Nachricht enthält den vset von y. Damit kann x seine eigene virtuelle Nachbarschaft berechnen. Bevor er sie zu seinem vset hinzufügt, sendet er an die neu entdeckten potenziellen virtuellen Nachbarn eine Setup-Request-Nachricht. Diese verfahren wie gerade beschrieben und senden an x eine Setup-Nachricht. Auf den Rückwegen werden neue Routen etabliert. Nach dem Erhalt der Nachrichten fügt x seine neuen virtuellen Nachbarn zum vset. Somit ist x dem Netzwerk beigetreten und wurde an der richtigen Stelle im Ring einsortiert.

3.5 Fehlerbehandlung

Wie gerade gesehen erkauft sich VRR die geringen Topologieänderungen beim Zugang mit höherer Komplexität: Mehrere Stufen des Zugangsverfahrens erhöhen seine Fehleranfälligkeit. Anzahl der Stufen wächst mit Kardinalität r. Es kann passieren, dass mehrere Knoten, im schlimmsten Fall mehrere zukünftige virtuelle Nachbarn, die dem Netzwerk gleichzeitig beitreten wollen. Da das Zugangsverfahren nicht atomar ist, werden sie sich gegenseitig stören: Sie können nacheinander an selben Knoten y eine Setup-Request-Nachricht schicken, ohne dem Netzwerk noch ganz beigetreten zu sein. Dieses kann die Nachbarschaftsbeziehungen bei y verändern. Es entsteht ein inkonsistenter Zustand. In diesem Fall wird y eine Setup-Fail-Nachricht an x zurückschicken. Diese wird dann den aktuellen vset von y enthalten, sodass x mit neuen Setup-Request-Nachricht erneut versuchen kann, dem Netzwerk beizutreten. Sollte es nach einer gewissen Zeit scheitern, müssen die bereits etablierten

aber nicht mehr gültigen Pfade aus dem Netzwerk entfernt werden. Dieses passiert übrigens auch, wenn ein virtueller Nachbar durch einen neuen Knoten aus dem vset verdrängt wird. Dabei wird eine Tear-Down-Path-Nachricht geschickt, die für das Herausnehmen des Pfades aus der Routingtabelle in den Knoten entlang des nicht mehr gültigen Pfades sorgt. Es kann passieren, dass das Netzwerk aufgrund der konkurrierenden Setup-, Setup-Request- und Tear-Down-Path-Nachrichten in mehrere eigenständige Netzwerke aufgeteilt wird. Wie wir das in der Einführung gesehen haben, hat ein Knotenausfall für proaktive Verfahren die Konsequenz, dass alle Pfade neu erkundet werden, für Reaktive wird der Pfad erst bei Bedarf festgestellt. Beide Ansätze haben zu unterschiedlichen Zeitpunkten das Fluten des Netzwerkes zur Folge. VRR hat einen Mechanismus, der ohne Fluten auskommt, die symmetrische Fehlererkennung. VRR bietet durch das Senden der HELLOs eine einfache Möglichkeit, ausgefallene Knoten aufzuspüren. Zusätzlich zu dem bereits besprochenen Inhalt eines HELLO, senden die Knoten eigene Sicht über jeden einzelnen Knoten in deren pset. Hat ein Nachbar so ein HELLO empfangen, weiß er, welche Sicht der Nachbar auf ihn hat. Für den Nachbarn kann er unbekannt sein, da eigene HELLOs bei ihm nicht ankommen etc. Aus diesen Informationen kann auf den Zustand der Verbindung zwischen den beiden geschlossen werden. Stellt Knoten x fest, dass die Verbindung zu seinem pset Nachbarn nicht mehr funktioniert, wird auch y das gleiche für x feststellen. Abbruch einer Verbindung kann auch durch das Fehlen von ACKs beim Senden von x nach y festgestellt werden. Jetzt werden sich beide gegenseitig aus den eigenen psets entfernten. Sollte eine virtuelle Route durch die gerade unterbrochene Verbindung geführt haben, können sowohl x als auch y, für die beiden Pfadhälften ein Tear-Down-Path durchführen, infolge wird der gesamte Pfad sauber entfernt und ein neuer kann mit Setup-Nachricht erkundet werden. Zu diesem gravierenden Schritt muss es nicht immer kommen. VRR bietet die Möglichkeit, lokale Pfadreparaturen im vset durchzuführen. Da die Pfade in VRR keinen Einschränkungen bezüglich der Länge (HOP) unterliegen, schafft es Voraussetzung um, wenn möglich, einen Pfad an der Stelle zur reparieren, wo der Fehler entstand, ohne die Notwendigkeit andere Knoten informieren zu müssen: Es werden nur die Knoten in der Umgebung des ausgefallenen Knotens bemüht. Um das sicherzustellen, wird die Routingtabelle (Tabelle 1) um ein Feld next¿next¿Start erweitert, also um einen weiteren Hop in Richtung Start. Auch die Setup-Nachricht wird verändert. Sie trägt jetzt die ID des jeweiligen Vorgängers mit sich, sodass dieses neue Feld gefüllt werden kann. Merkt x jetzt einen Knotenausfall von y, schaut es sich die betroffenen vset Routen, indem es versucht Alternativen für next¿Start in der Nachbarschaft zu finden. Ist Start ein pset Nachbar, dann wird es für next¿Start eingetragen, die Route wurde gar verkürzt und x ist fertig. Sonst wird als nächstes untersucht, ob next¿next¿Start ein Nachbar im pset ist. Wenn ja wird next¿Start überbrückt. Schlägt das fehl, sucht jetzt x in seiner eigenen Nachbarschaft nach einem Knoten, der selbst Nachbar von next¿next¿Start ist. Ist die Suche erfolgreich, kann auch hier lokal repariert werden. Informationen dafür bekommt er aus den ausgetauschten HELLOs. Wie zu sehen war, greift VRR beim Ausfall eines Pfades nur auf lokal gespeicherte Informationen zurück, es müssen keine zusätzlichen Nachrichten verschickt werden, weder um die Überbrückung herauszufinden, noch um diese Änderung der Routen anderen Knoten mitzuteilen.

3.6 Partitionierung des Netzwerks

VRR bietet einen Mechanismus an, um kleinere Netzwerke zu einem zusammenzuführen, z. B. nach einem gerade beschriebenen Fehlerfall. Zu diesem Zweck werden aus jedem Teilnetzwerk Repräsentanten festgestellt. Ein Repräsentant ist ein Knoten, der am dichtesten an null liest. Der Repräsentant kann das selbst feststellen. Stellt jetzt ein beliebiger Knoten fest, dass er einen Repräsentanten kennt (initial ist das der physikalische Nachbar), baut er zu ihm eine virtuelle Route auf, und speichert sie in seiner Routingtabelle ab. Diese Routen zu den Repräsentanten werden in den HELLOs vom Knoten zu Knoten weitergegeben, sodass sie sich über die Grenzen der Teilnetzwerke ausbreiten werden, und so einen anderen Repräsentanten erreichen. Dadurch können die beiden nach dem bereits besprochenen Mechanismus sich selbst als virtuelle Nachbarn erkennen, und so die beiden Netzwerke vereinen. VRR optimiert diesen Prozess, um Overhead zu vermeiden. Die Informationen über die Repräsentanten in HELLOs haben einen Zähler, der vom Repräsentanten selbst erhöht wird. Sollten bei einem Knoten keine neuen Informationen (mit erhöhtem Zähler) eintreffen, werden sie auch nicht mehr propagiert. Das kann z. B. passieren, wenn der bisherige Repräsentant nach der Vereinigung der Netze diese Rolle nicht mehr innehält. Außerdem merken sich die Knoten nur die aktuellsten Routen zu zwei Repräsentanten, die der Null am dichtesten sind. Das beschriebene Verfahren benötigt keine zusätzlichen Kontrollnachrichten - Overhead. Es muss lediglich HELLO um wenige Daten erweitert werden. Aufgrund der Zeitabstände beim Versenden der HELLOs kann sich dieses Verfahren länger hinziehen. Es gibt, wie wir es bei der Zugangsfehlerbehandlung gesehen haben, keine Garantie, dass die Vereinigung beim ersten Versuch geschehen wird. Solange die Teilnetzwerke sich physikalisch berühren, werden neue Vereinigungsversuche unternommen.

4 Evaluierung

VRR kann hier gegenüber der Konkurrenz in mehreren verschiedenen Simulationen mit guter Performance überzeugen, das gute Bild setzt sich auch in der realen Umgebung fort. Ausgiebig getestet wurde VRR in der simulierten Umgebung ns-2[**?**]. Es wurden mehrere Testfelder simuliert, die verschiedene Aspekte einer Maschen-Umgebung untersucht. Die Kandidaten wurden dabei stets anhand Anzahl der korrekt abgelieferten Pakete, dem End-to-end-Delay und der Anzahl der gesendeten Kontrollnachrichten beurteilt. In den simulierten Umgebungen wurde eine feste Knotendichte auf einem Quadratmeter veranschlagt. Bei einem Versuch mit mehr Knoten wurde auch die Fläche erhöht, um die Dichte beizubehalten. Die Knotenanzahl bewegte sich zwischen 25 und 200. Es wurde die Performance sowohl im statischen Szenario, als auch in einem Beweglichen untersucht. Dabei bewegten sich die Knoten in wahllose Richtungen mit einer konstanten Geschwindigkeit. Die Protokolle hatten 1000s für den Netzaufbau. Danach wurde für 900s Messung durchgeführt. VRR war bei 200 Knoten schon nach 24.3s fertig und hatte im Durchschnitt 110.4 Nachrichten/Knoten verschickt. Ein einmaliges Fluten benötigt schon 200 Nachrichten. Die Kandidaten neben VRR waren: Reaktive Protokolle DSR und AODV und das proaktive DSDV[**?**]. 1. DSR: Routen werden vor dem Versenden herausgefunden. Nachrichten haben Routen im Header. Alternative Routen sind bekannt. DSR hat Probleme mit dem Erkennen veralteter Routen. Caching der Routen. 2. AODV: Basiert auf DSDV: Routen werden nur bei Bedarf erkundet. Jeweils nur eine Route/Ziel ist bekannt. 3. DSDV: Jeder Knoten speichert alle möglichen Ziele

in der Setup-Phase. Bei VRR wurde die Kardinalität auf 4 gesetzt. Zeitabstand für HELLOs wurde auf 1s festgelegt. IDs hatten die Länge von 4 Bytes. Ein Szenario war, das Verhalten bei steigender Netzauslastung zu erfahren, Bei konstant 100 Knoten. Es wurden UDP Pakete mit konstanter Bitrate CBR verschickt. Angefangen mit 1 bis 200 Übertragungen auf die Knoten verteilt. Es wurde sichergestellt, dass ein Knoten max. 2-mal Quelle der Übertragung war. Ziele waren willkürlich gewählt. Im statischen Szenario liefern alle Protokolle die Pakete fehlerfrei ab. Erst ab ca. 150 Sendungen fängt die Zuverlässigkeit an einzubrechen. Bei 200 Sendungen schneidet DSR am besten ab mit ca. 85Auch weitere Simulationen bestätigen die gerade beschriebenen Ergebnisse. VRR besitzt in allen Szenarios im Vergleich zu anderen Protokollen ein sehr kurzes End-to-end Delay. Auch das Verhältnis korrekt abgelieferter Pakete geht fast immer zugunsten von VRR aus. Der Overhead/Nachricht Anteil ist bei VRR meist am geringsten, manchmal ist der Abstand zu anderen Protokollen sehr groß. Die Beurteilung der Mitstreiter in diesen Simulationen fällt durchwachsen aus. Einige Protokolle können unter bestimmten, auf sie zugeschnittenen Bedingungen punkten, unter anderen liegen sie hinten. Über alle Simulationen, unabhängig ob statisch oder mobil schneidet VRR konstant gut ab. Das reale Testfeld setzte sich aus einem Netzwerk von 30 Knoten in 802.11a Umgebung. VRR hatte hier nur einen Konkurrenten: MR-LQSR. Es ist ein eine Weiterentwicklung von dem reaktiven DSR. MR-LQSR zieht für die Routenwahl das bereits angesprochene ETX und wird zusammen mit dem von Microsoft ausgelieferten Mash-Connectivity-Layer MCL[**?**] Framework eingesetzt. Für den Versuch wurde in einer parallelen Testumgebung MR-LQSR durch VRR ersetzt. Das MLC-Framework bietet den oberen Protokollschichten eine transparente Schnittstelle, sodass TCP eingesetzt werden kann. Die Testumgebung bestand aus 30 PCs mit Windows XP auf ein Stockwerk verteilt. Der Durchmesser einer Sendezelle betrug 4. Für VRR wurde die Zeit für HELLOs auf 2s und r auf 4 gesetzt. Als IDs wurden 48-Bit-MAC-Adressen benutzt. Die Messungen zeigten, dass VRR auch unter den realen Bedingungen punktet. In einem Test erreichte VRR in 70

5 Zusammenfassung

VRR zeigt in der Tat fortschrittliche Mechanismen beim Routing. Die Anleihen von DHT lassen es zu, die bisher benutzten Mechanismen über Bord zu werfen, ohne Einbußen bei der Performance, ganz im Gegenteil, wie die Ergebnisse der Evaluierung es zeigen. VRR schafft es bei den eintretenden Topologieänderungen die negative Auswirkung, nämlich Bekanntgeben der neuen Routen zu vermeiden. Auf den zweiten Blick entstehen gerade dadurch sehr komplexe Seiteneffekte, die wie im Abschnitt zur Fehlerbehandlung beschrieben wurde, zur Partitionierung des Netzwerkes führen können. VRR ist komplex geworden, aber immer noch beherrschbar, wie die Performance-Ergebnisse es zeigen. Die Fähigkeit fast beliebige Ids (IP-Adressen) zu verwenden stellt sicher, dass VRR sehr gut mir mit TCP/IP, also nach oben hin in dem Protokollstack kooperieren wird, da IP im Übrigen sehr genügsam ist. Nach unten hin stellt die Unfähigkeit für Multicast eine Hürde für das ARP dar. Dessen Funktionalität muss durch die HELLOs nachgebildet werden. Als Alternative lassen sich sicherlich MAC-Adressen benutzen, was allerdings den Einsatz sehr einschränkt. Insgesamt ist es wünschenswert, dass dieses junge Protokoll eine Verbreitung findet.

Literatur

[AD HOC] Ad hoc Netze http://de.wikipedia.org/wiki/Ad$_{h}oc$20.01.2007

[MTR] R. Draves, J. Padhye, and B. Zill. Comparison of routing metrics for static multi-hop wireless networks. In SIGCOMM, August 2004.

[ETX] Berechnung des ETX Wertes http://wiki.leipzig.freifunk.net/ETX-Wert 20.01.2007

[VRR] M. Caesar, M. Castro, E. Nightingale, G. O'Shea, A. Rowstron. Virtual Ring Routing: Network Routing Inspired by DHTs, ACM SIGCOMM 2006

[NS-2] Network Simulator NS-2 http://www.isi.edu/nsnam/ns/ 20.01.2007

[VGL] Gegenüberstellung der Protokolle: DSR und AODV DSDV. http://www-md.e-technik.uni-rostock.de/ma/hm13/lehre/v$_{a}dhoc.ppt$20.01.2007

[MSH] Mash-Connectivity-Layer http://research.microsoft.com/mesh/ 20.01.2007