

**Mikroprotokoll-basierte Komposition
von Routingprotokollen
am Beispiel von AODV und NXP/MPR**

I. Fliege, A. Geraldys

Technischer Bericht 344/05

**Mikroprotokoll-basierte Komposition
von Routingprotokollen
am Beispiel von AODV und NXP/MPR**

I. Fliege, A. Gerald

Fachbereich Informatik, TU Kaiserslautern, Kaiserslautern
{fliege,gerald}@informatik.uni-kl.de

Technischer Bericht 344/05

Fachbereich Informatik
TU Kaiserslautern
Postfach 3049
67653 Kaiserslautern

Mikroprotokoll-basierte Komposition von Routingprotokollen am Beispiel von AODV und NXP/MPR

Ingmar Fliege, Alexander Gerald
{fliege,geraldy}@informatik.uni-kl.de

AG Vernetzte Systeme
TU Kaiserslautern

21.11.2005

Zusammenfassung

Mobile Ad-hoc-Netzwerke stellen eine große Herausforderung bei der Entwicklung von Netzwerkprotokollen da. Diese Netzwerke aus gleichberechtigten Knoten bilden sich spontan und ermöglichen ohne weitere Infrastruktur eine Multi-Hop-Kommunikation. Dabei bereiten die Eigenschaften von Funknetzen, die Mobilität der Knoten und die eingeschränkte Ressourcensituation besondere Probleme, welche bei der Entwicklung berücksichtigt werden müssen und ein Maßschneidern von Protokollen und Anwendungen erfordern.

In diesem Beitrag wird eine generische Struktur von Routingprotokollen vorgestellt, die eine komponentenorientierte Entwicklung von Protokollen ermöglicht. Zum Einsatz kommen dabei Mikroprotokolle, d.h. sofort verwendbare, abgeschlossene Funktionalitäten, die durch Komposition ein vollständiges Protokoll beschreiben.

Am Beispiel von AODV und NXP/MPR wird das systematische Vorgehen zur Komposition von Routingprotokollen vorgestellt. Dabei wird der Kontrollfluss zwischen den einzelnen Mikroprotokollen modelliert, ähnliche Funktionalitäten ersetzt und redundante Komponenten entfernt. Mit der Komposition entsteht ein maßgeschneidertes, höherwertiges Protokoll.

1 Einführung

Mobile Ad-hoc-Netze stellen eine Vision zukünftiger Kommunikations- und Datennetze dar. Sie verzichten auf teure und aufwändige Infrastruktur und können damit zu jedem Zeitpunkt und an jedem Ort spontan aufgebaut werden. Außerdem unterstützen diese Netze ein Höchstmaß an Mobilität; sie sind damit in der

Lage, auf topologische Änderungen durch Bewegung der Netzwerkknoten oder auf Änderungen in der Anzahl der beteiligten Knoten zu reagieren. Um dies zu ermöglichen, geht man meist davon aus, dass jeder Knoten bei Bedarf die Aufgaben der nicht vorhandenen Infrastruktur ganz oder teilweise übernimmt, z.B. indem jeder Knoten einen Beitrag zum Routing leistet.

Darüber hinaus stellen sich Ad-hoc-Netze einer Vielzahl neuer Herausforderungen. Verteilte Anwendungen und Protokolle spielen eine wesentlich stärkere Rolle als dies in Infrastrukturnetzen (wie z.B. in GSM-Netzen, Lokalen Ethernets oder im Internet) der Fall ist. Außerdem stellen die technologisch gegebenen Einschränkungen, wie z.B. die eingeschränkte Verbindungsqualität zwischen den Knoten, die Kurzlebigkeit der Kommunikationsbeziehungen sowie Probleme durch die Funkübertragung (keine dedizierten "Leitungen", Kollisionen, Hidden-Station, Exposed-Node) eine große Rolle.

Die Netzwerkknoten bestehen meist aus kleinen, eingebetteten Computern oder Mikrocontrollern mit harten Ressourcenbeschränkungen (Energie, Rechenleistung, Speichergröße, etc.). Diese Ressourcenbeschränkung sowie die technischen Problemstellungen der Ad Hoc Netze müssen bei der Spezifikation neuer Anwendungen und Protokolle berücksichtigt werden und verhindern den Einsatz allgemein anwendbarer Protokolle. Stattdessen müssen Protokolle entwickelt werden, die auf den Ad-hoc-Kommunikationsbereich, spezielle Hardware oder einzelne Anwendungen in diesem Bereich maßgeschneidert sind.

Dieses Maßschneidern ist ein aufwändiger Vorgang, da beispielsweise in der kurzen Zeit der Forschung im Bereich der Ad-hoc-Netze eine Vielzahl an Routingprotokollen entwickelt werden mussten, die jedes Mal von Grund auf neu spezifiziert wurden. Hierbei wird oft nicht berücksichtigt, dass jedes weitere Protokoll zu großen Teilen als eine Komposition von Komponenten bisheriger Protokolle gesehen werden kann. Diese komponentenzentrierte Sicht könnte dabei helfen, die Unterschiede zu existierenden Verfahren präzise hervorzuheben, bereits entwickelte Teile wiederzuverwenden und durch den Griff zu bereits getesteten und bewährten Bausteinen auch die Qualität von Protokollspezifikationen zu erhöhen.

Wir werden in diesem Bericht zeigen, wie ein neues Routingprotokoll für Ad-hoc-Netze aus Komponenten zweier anderer Routingprotokolle aufgebaut werden kann, wodurch die Entwurfskomplexität drastisch reduziert werden kann. Wir werden hierbei besonders auf die Methodologie eingehen.

2 Ad-hoc-Routingprotokolle

Die Aufgabe eines Routingprotokolls liegt in der Berechnung eines Leitwegs von einem Knoten, der Daten versenden möchte, zu einem oder mehreren Empfängern, sowie in der Weiterleitung von Paketen über den berechneten Leitweg. Wird nur ein Zielknoten adressiert, so spricht man von Unicast-Routing, eine Gruppe wird im Multicast adressiert, bei Broadcast-Routing sollen alle Knoten eines Netzes erreicht werden – weitere Kommunikationsformen wie Anycast, Geocast oder Concast klammern wir in diesem Bericht aus. Unabhängig

davon, welche Adressierung gewählt wird, muss Ad-hoc-Routing grundsätzlich dynamisch (adaptiv) ausgebildet sein, eine statische Konfiguration kommt aufgrund der Veränderlichkeit des Netzwerks durch Mobilität nicht in Frage. Die Ermittlung des (sich ändernden) Netzzustands kann nun entweder reaktiv, d.h. bei Bedarf einer Route, proaktiv, d.h. im Voraus, oder hybrid als Mischform erfolgen. Die meisten Verfahren wählen dabei eine Route nach topologischen Aspekten (Knoten und Links zwischen ihnen) aus, einige setzen auf einen positionsbasierten Ansatz, der geographische oder lokale Koordinaten nutzt, um Leitwege zu finden. Die Sicht des Netzzustands kann hierbei entweder auf lokale Informationen beschränkt sein, globale Informationen (d.h. das komplette Netz) umfassen oder Teile davon (partiell). Die Informationen können außerdem vollständig vorliegen oder aggregiert (zusammengefasst). In der Kategorie der Verfahren mit globalen, aggregierten Informationen liegt beispielsweise Distance-Vector-Routing, während Link-State-Routing globale, vollständige Informationen benötigt. Wenn die Auswahl eines Leitwegs Dienstgütereorderungen unterliegt (z.B. Bandbreite oder Delay), sprechen wir von Dienstgüterouting, andernfalls sprechen wir von Best-Effort-Routing.

Die Verwendung der Leitwege kann nun geschehen, indem der Absender einen kompletten Leitweg vorgibt (quellbasiert), indem der genutzte Leitweg nur hop-by-hop festgelegt wird (verteilt), oder indem hierarchisch auf verschiedene Ebenen eine unterschiedliche Sicht auf das Netz ausgenutzt wird. In diesem mehrdimensionalen Raum liegen eine Vielzahl bisher spezifizierter Routingprotokolle für Ad-hoc-Netze, beispielsweise:

AODV:

Ad Hoc On Demand Distance Vector Routing [4]: Unicast, reaktiv, topologiebasiert, partiell, aggregiert, best-effort, verteilt.

DSDV:

Destination Sequenced Distance Vector Routing [9]: Unicast, proaktiv, topologiebasiert, global, aggregiert, best-effort, verteilt.

DSR:

Dynamic Source Routing (DSR [11]: Unicast, reaktiv, topologiebasiert, partiell, aggregiert, best-effort, quellbasiert.

ODMRP:

On Demand Multicast Routing Protocol [10]: Multicast, reaktiv, topologiebasiert, partiell, aggregiert, best-effort, verteilt.

Fluten:

Basisfluten: Broadcast, reaktiv, topologiebasiert, lokal, best-effort, verteilt.

NXP/MPR:

Neighbor Exchange Protocol with Multi Point Relay [5]: Broadcast, proaktiv, topologiebasiert, partiell, vollständig, best-effort, verteilt.

3 Modulare Protokollentwicklung

In der Forschung existieren verschiedene Ansätze, um eine modulatorientierte Protokollentwicklung vorzunehmen. Ein Vertreter ist das OSI-Referenzmodell, das eine Unterteilung von Protokollen in verschiedene Schichten vornimmt. Auch bei andere Ansätzen, wie DaCaPo [2] und ADAPTIVE [3], bestehen die Module aus einzelnen Protokollschichten. Im Gegensatz dazu sind *Mikroprotokolle* [1] feingranulare Bausteine zur Entwicklung von Kommunikationsprotokollen, die ohne a priori Festlegung einer speziellen Protokollarchitektur und daher sehr flexibel eingesetzt werden können.

Mikroprotokolle stellen einen Protokollbaustein mit einer abgeschlossenen, sofort verwendbaren Funktionalität da, der über eine wohldefinierte Schnittstelle verfügt. Durch die Komposition dieser Bausteine lassen sich komplexe, maßgeschneiderte Kommunikationsprotokolle entwickeln. Dabei wird durch die Wiederverwendung der Entwicklungsaufwand deutlich reduziert und zusätzlich die Qualität gesteigert, da auf bereits verwendete und getestete Bausteine zurückgegriffen wird.

Die Granularität von Mikroprotokollen lässt sich nur im Zusammenhang mit einer spezifischen Funktionalität beantworten. Beispielsweise verfügen Protokolle der Sicherungsschicht (LLC) über größtenteils einfache reaktive Funktionalitäten, wie das Berechnen von Prüfsummen, das Quittieren oder eine Flusskontrolle. Im Gegensatz dazu findet man in Routingprotokollen größere und komplexere Teilfunktionalitäten, welche verschränkte Zugriffe auf Tabellen durchführen, dabei aufwändige Berechnungen vornehmen und verschiedene Time-outs behandeln müssen.

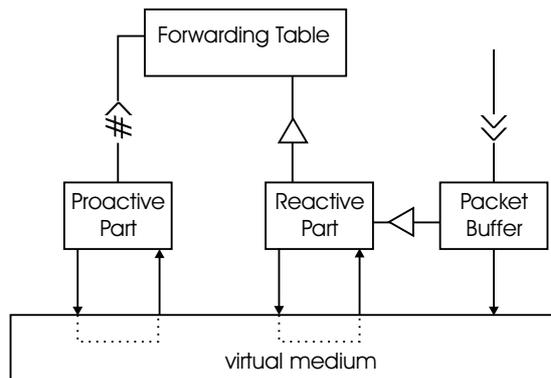


Abbildung 1: Generische Architektur von Routingprotokollen

In Abbildung 1 sieht man eine vereinfachte generische Struktur, die sich aus der Untersuchung verschiedener Routingprotokolle ergeben hat. Wir identifizieren einerseits den **Packet Buffer**, der zu verschickende Datenpakete erhält und dabei, durch den Operator $\rightarrow\rightarrow$ angestoßen, versucht, über den **Reactive Part** an Informationen zur Weiterleitung (bspw. den nächsten Hop) zu

gelangen. Dies geschieht über den gerichteten Operator \dashv , der einen Aufruf mit dazugehöriger Antwort darstellt. Der reaktive Teil des Routing übernimmt eine Koordinationsrolle zwischen der aktuellen **Forwarding Table**, die eben diese Informationen beinhalten kann, und dem **Packet Buffer**. Der reaktive Teil kommuniziert bei Bedarf mit dem reaktiven Teil anderer Knoten über das virtuelle Medium (dargestellt mit den Operatoren \dashv).

Falls das Protokoll einen **Proactive Part** beinhaltet, kommuniziert dieser mit dem proaktiven Teil der Nachbarknoten ebenfalls über das virtuelle Medium. Die Ergebnisse des proaktiven Teils werden ggfls. zusammengefasst und in die Forwarding Table geschrieben (Operator $\#>$). Die Forwarding Table enthält letztlich nur die Informationen, die für die Weiterleitung von Paketen notwendig sind.

4 AODV und NXP/MPR

4.1 AODV

Das Ad hoc On Demand Distance Vector Routing (AODV, [4]) ist eine spezielle Ausprägung des Distance Vector Routings. Routen werden nur reaktiv (bei Bedarf) gesucht, indem der Quellknoten einen Route-Request (RREQ) sendet, der u.a. die ID der Quelle, des Ziels sowie eine Sequenznummer beinhaltet. Jeder Empfänger dieses RREQ überprüft nun ob er bereits eine Route zum Ziel kennt. Falls ja, wird ein Route Reply erzeugt (RREP), ansonsten wird ebenfalls ein RREQ gesendet. Dieser RREQ zieht sich also wie eine Flutwelle durch das komplette Netz und erreicht so – falls möglich – den Zielknoten oder einen Knoten, der bereits eine Route zum Zielknoten kennt. Eine Schleifenbildung der RREQs wird unterbunden durch das lokale Caching des eindeutigen Quelle/Sequenznummer-Paars sowie einen Time-to-live-Wert, der für jeden Hop dekrementiert wird. Die RREPs suchen den Rückweg durch die vorhergehenden Knoten, die sich jeder Knoten in der RREQ-Welle gespeichert hat. Gleichzeitig werden die RREQs aggregiert, indem Knoten, die mehrere RREPs empfangen, die nachfolgenden RREPs mit höheren Kostenwerten (Hop-Zahl bis zum Ziel) ignorieren.

Die Aktualität der Routen wird dadurch gewährleistet, dass Routen erst gesucht werden, wenn ein Bedarf besteht, nur gecached werden, solange ein Bedarf besteht, und die Korrektheit der Route bei bestehender Verbindung durch Link-Layer-Maßnahmen oder Hello-Pakete überwacht wird. Da der von uns verwendete Link-Layer, bzw. das Interface zu diesem [12], keine Signalisierung für fehlerhaft übertragene Datenpakete (z.B. durch ausbleibende WLAN-Acknowledgements) unterstützt, greifen wir auf die Hello-Version von AODV zurück. Hier wird in regelmäßigen Abständen ein Hello-Paket gesendet, um Nachbarknoten darüber zu informieren, dass noch eine Verbindung zwischen den beiden Knoten bestehen kann. Sobald ein Hello-Paket mehrfach ausfällt, reagiert der dies erkennende Knoten mit einer erneuten Routensuche, da der Bedarf an der Route noch besteht (sonst wäre die Route nicht mehr in der

Tabelle).

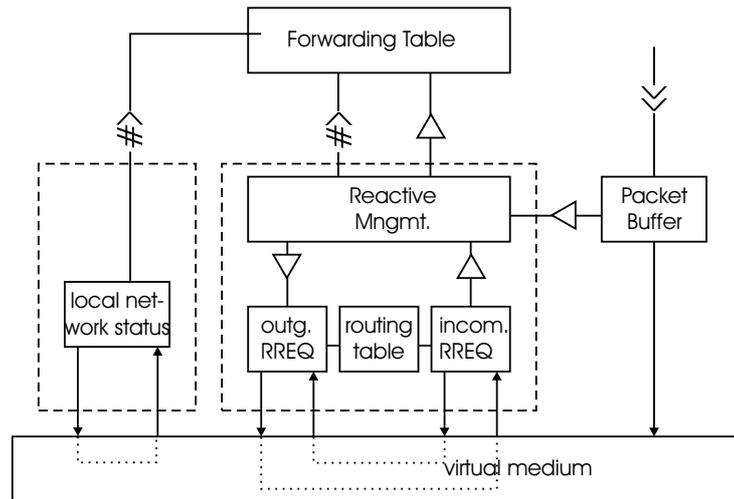


Abbildung 2: Architektur von AODV

Die Einordnung von AODV in das generische Modell aus Abb. 1 ist in Abb. 2 zu sehen. Der proaktive Anteil von AODV besteht aus der Erfassung der 1hop-Nachbarschaft durch das Hello-Protokoll. Der reaktive Teil ist aufgeteilt in eine Management-Komponente und zwei RREQ-Einheiten, die für den RREQ-RREP-Handshake (lokal gestartet und remote gestartet) zuständig sind. Beide müssen auf Routing-Informationen zugreifen, die zu detailliert für die Forwarding Table sind, daher ist ihnen direkt eine erweiterte Routingtabelle zugeordnet, die über das reaktive Management mit der Forwarding-Table bei Bedarf synchronisiert wird.

4.2 NXP/MPR

NXP/MPR [5] ist eine Optimierung des einfachen Flutens. Beim einfachen Fluten sendet jeder Rechner, der ein neues Paket erhält, dieses weiter. Dies bedeutet, dass in einem Netzwerk mit n Knoten das Senden einer Ursprungsnachricht $n - 1$ Wiederholungen hervorruft. In größeren Netzen, insbesondere wenn das Netz dicht ist, d.h. sehr viele Knoten auf einem engen Raum positioniert sind, führt dies zu Bandbreiteengpässen und dem Broadcast-Storm-Problem [8]. Einige Versuche wurden unternommen, um die Anzahl der Forwarder und damit auch die Gesamtnetzlast zu reduzieren, beispielsweise probabilistische Ansätze. Bei NXP/MPR überwacht jeder Knoten die 2hop-Nachbarschaft (mittels Neighbor Exchange Protocol, NXP) und bestimmt darauf basierend, welcher 1hop-Knoten Forwarder für diesen Knoten sein soll, um 2hop-Nachbarn abzudecken und welche 1hop-Nachbarn nicht benötigt werden (Multi Point Relay Protocol, MPR). Wir berechnen also eine Forwardergruppe pro letztem Hop der Nach-

richt.

Mit Hilfe des NXP wird zunächst nur die aktuelle Sequenznummer des Knotens übertragen. Nach kurzer Zeit kennt jeder Knoten hierüber seine Nachbarknoten. Die Liste der Nachbarknoten wird nun ebenfalls über NXP übertragen, so dass anschließend jeder Knoten seine 2hop-Nachbarn kennt – nämlich die 1hop-Nachbarn der 1hop-Nachbarn, die wiederum nicht schon direkte Nachbarn des Knotens sind. Das NXP-Verfahren ermittelt also proaktiv eine vollständige Beschreibung der 2hop-Nachbarschaft eines Knotens.

Darauf basierend ermittelt MPR eine Menge von Forwardern aus der 1hop-Nachbarschaft, die zusammengenommen alle 2hop-Nachbarn abdecken. Diese Entscheidung wird schließlich proaktiv vom MPR-Protokoll an alle 1hop-Nachbarn übertragen, diese speichern die Entscheidung in der Forwarding Table und nutzen sie bei der nächsten eintreffenden Nachricht des jeweiligen Senders.

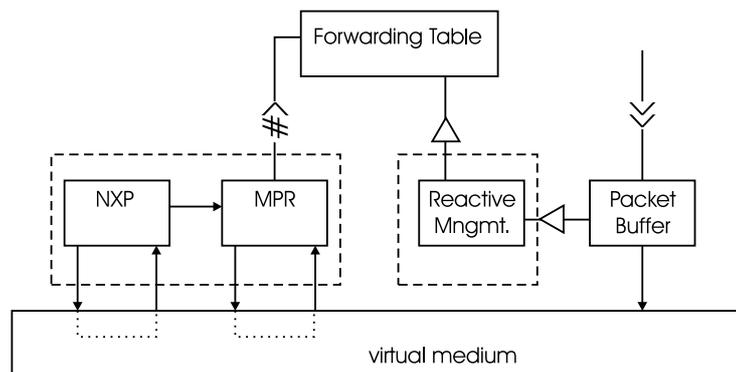


Abbildung 3: Architektur von NXP/MPR

Wir können folglich NXP/MPR in die Struktur des generischen Routing integrieren, wie dies in Abbildung 3 gezeigt ist. Der proaktive Anteil des Protokolls ist stark ausgeprägt, während der reaktive Teil nur Managementaufgaben erledigt und keine direkte Kommunikation über das Netzwerk durchführt.

5 Komposition von Routingprotokollen

Mit der Komposition von NXP/MPR und AODV wollen wir erreichen, dass der Overhead von AODV bei Route Requests gemildert wird. AODV erzeugt aufgrund seines reaktiven Verhaltens zwar eine sehr niedrige Grundlast (lediglich bestehend aus den Hello-Nachrichten); sobald eine Route verlangt wird, erfolgt jedoch ein Broadcast-Sturm, der zu einer (kurzfristigen) Überlast im Netz führt. Durch die Verwendung von NXP/MPR für die RREQs erhoffen wir eine Vermeidung des Broadcast-Sturm bei gleichzeitiger Aufrechterhaltung der Funktionalität und weitestgehender Aufrechterhaltung der Effizienz von AODV.

Den weitaus größeren Eingriff stellt ein Optimierungsschritt dar, der aus NXP/MPR alle Teile der Spezifikation entfernt, die nach der Komposition nicht mehr notwendig sind. Hierzu gehören Teile des Muxers/Demuxers sowie die nutzdatenverarbeitenden Teile des MPR-Protokolls. Diese werden nicht mehr genutzt, da die Nutzdatenverarbeitung und die Nutzung der MPR-Entscheidungen in AODV integriert wurden.

Nach diesem Schritt ist die erfolgreiche Komposition anhand von Simulationen zu prüfen. Die Routensuche von AODV muss noch funktionieren und sie muss ohne den Broadcast-Sturm verlaufen. Eine Analyse der durchgeführten Simulation zeigt jedoch, dass die noch vorliegende Überschneidung zweier Hello-Protokolle (das AODV-Hello und das NXP) zu einer unnötigen Erhöhung der Netzlast führen.

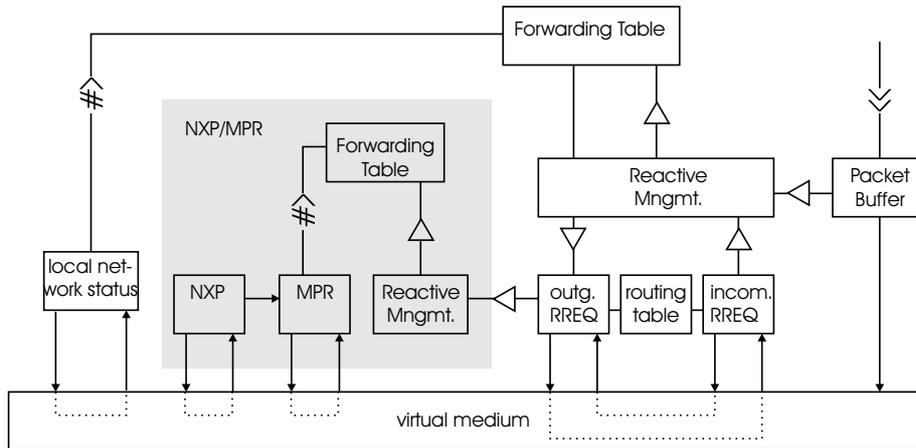


Abbildung 5: Komposition von NXP/MPR und AODV – Schritt 2

5.3 Zusammenfassung der Hello-Prozesse

Im letzten Schritt können nun die beiden Hello-Protokolle zusammengefasst werden (s. Abb. 6). NXP ermittelt die vollständige 2hop-Nachbarschaft eines Knotens, während AODV lediglich 1hop-Nachbarn benötigt, um die Lebendigkeit von Routen zu überwachen. Aus diesem Grund binden wir AODV an das NXP an und nutzen Informationen aus, die NXP bereits besitzt. Bei diesem Schritt wird automatisch eine gemeinsame Hello-Periode geschaffen und die Effizienz des Gesamtprotokolls durch den Wegfall redundanter (und nicht-synchronisierter) Nachrichten gesteigert.

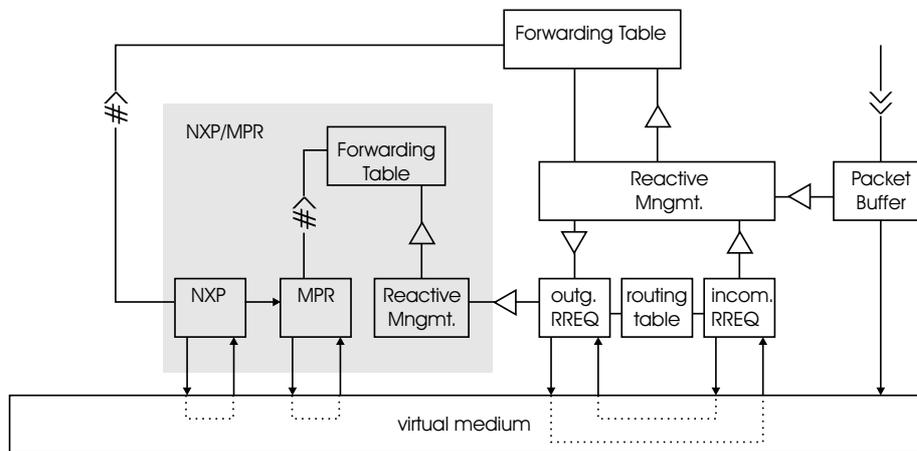


Abbildung 6: Komposition von NXP/MPR und AODV – Schritt 3

6 Zusammenfassung

Wir haben anhand des AODV+NXP/MPR-Beispiels gezeigt, dass die Zerlegung von Protokollen in Mikroprotokolle und die anschließende Komposition der Mikroprotokolle einen schnellen Weg zu komplexen, maßgeschneiderten bzw. optimierten Protokollen darstellt. Gleichzeitig bietet sich dem Entwickler eine neue Abstraktionsebene, die sich auf Mikroprotokollebene auf abgeschlossene Funktionalitäten stützt und damit eine Simplifizierung des Designprozesses darstellt. Bezüglich des Beispiels von AODV und NXP/MPR stehen weitergehende Performanzanalysen noch aus. Aufgrund dieser zukünftigen Ergebnisse werden mögliche (Parameter-)Anpassungen und Optimierungen diskutiert werden können.

Desweiteren wird der Ausbau der Mikroprotokollbibliothek einen hohen Stellenwert einnehmen, erst durch ein möglichst hohe Verfügbarkeit vielfältiger, ausgereifter Mikroprotokolle kann die zukünftige Entwicklung neuer Protokolle bestmöglich unterstützt werden.

Literatur

- [1] I. Fliege, A. Gerald, R. Gotzhein, P. Schaible: A Flexible Micro Protocol Framework, in: 4th SAM (SDL And MSC) Workshop, 2004.
- [2] M. Vogt, T. Plagemann, B. Plattner, T. Walter: A Run-time Environment for DaCaPo, in: Proc. of INET'93, Int. Networking Conf. Internet Society, San Francisco, 1993.
- [3] D. Schmidt, D. Box, T. Suda, ADAPTIVE: A flexible and adaptive transport system architecture to support lightweight protocols for multimedia

applications on high speed networks, in Proceedings of the Symposium on High Performance Distributed Computing Conference, Amsterdam, 1992.

- [4] C. Perkins, E. Belding-Royer, S. Das: Ad Hoc On Demand Distance Vector (AODV) Routing, RFC 3561, <http://www.ietf.org/rfc/rfc3561.txt>. 2003.
- [5] T. Sonntag: Optimiertes Fluten am Beispiel einer Sprechfunkanwendung, Projektarbeit, Technische Universität Kaiserslautern, Fachbereich Informatik, 2005.
- [6] F. Rothländer: DSDV/AODV – Analyse, Spezifikation und Simulation, Diplomarbeit, Technische Universität Kaiserslautern, Fachbereich Informatik, 2005.
- [7] A.Gerald, R. Gotzhein: Micro Protocol Based Development of Mobile Ad Hoc Networks, Personal Wireless Communications (PWC 2003), Venice/Italy, 2003.
- [8] S. Y. Ni, Y. C. Tseng, Y. S. Chen, J. P. Sheu: The Broadcast Storm Problem in MANETs. In: Proc. International Conference on Mobile Computing and Networking (MobiCOM), 1999.
- [9] Charles E. Perkins, Pravin Bhagwat: Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. In: Proceedings of the SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications, 1994.
- [10] S.-J. Lee, W. Su, and M. Gerla: On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks; ACM/Baltzer Mobile Networks and Applications, special issue on Multipoint Communication in Wireless Mobile Networks, 2000.
- [11] David B. Johnson, David A. Maltz, Yih-Chun Hu: The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). In: Internet Draft, draftietf-manet-dsr-09.txt, 2004.
- [12] I. Fliege, A. Gerald, S. Jung, T. Kuhn, C. Weibel, C. Weber: Konzept und Struktur des SDL Environment Frameworks – SEnF, Technischer Bericht 341/05, Fachbereich Informatik, Technische Universität Kaiserslautern, 2005.