

Kooperative schichtenübergreifende Überlastkontrolle für mobile Ad-Hoc-Netzwerke

Björn Scheuermann, Christian Lochert, Martin Mauve

Institut für Informatik, Heinrich-Heine-Universität Düsseldorf

Zusammenfassung Die oft beobachteten Schwierigkeiten beim Einsatz gängiger und – in anderen Umgebungen – lange bewährter Netzwerkprotokolle in drahtlosen Multihop-Netzwerken sind eine direkte Folge der besonderen Eigenschaften dieser speziellen Umgebung. Bisherige Arbeiten haben diese Eigenschaften in der Regel als Nachteil bzw. ausschließlich als Quelle von Problemen angesehen. Hier werden die Besonderheiten der Umgebung hingegen aus einem anderen Blickwinkel betrachtet und konstruktiv als Grundlage für alternative Ansätze genutzt. Dabei zeigt sich, dass gerade diese Eigenschaften häufig als Fundament für unkonventionelle und gleichzeitig sehr effektive Lösungen dienen können.

1. Einführung

Kooperative drahtlose Multihop-Netzwerke erlauben ihren Nutzern die Kommunikation ohne feste Infrastruktur auch deutlich über die Funkreichweite eines einzelnen Gerätes hinaus. Die Einsatzmöglichkeiten solcher Netzwerke unterscheiden sich dabei ebenso wie die Fähigkeiten der eingesetzten Geräte: Die Spanne reicht von ressourcenarmen Sensorknoten, die ausschließlich und speziell für das Erfassen, Weiterleiten und Versenden von Messungen gebaut sind, bis hin zu durchaus leistungsstarken Systemen mit umfangreichen Speicher- und Prozessorkapazitäten.

Diskutiert werden als Einsatzgebiete von leistungsstarken Geräten beispielsweise die Vernetzung von Fahrzeugen auf der Straße zur Verbesserung von Verkehrssicherheit und Fahrkomfort, Mesh-Netzwerke, die die Reichweite von Internetzugangspunkten vergrößern, oder die Kommunikation in Katastrophenfällen, wenn keine Infrastruktur verfügbar ist. Bei den eher leistungsschwächeren Geräten stehen Überwachungs-, Kontroll- und Alarmierungsanwendungen im Mittelpunkt der Überlegungen. Gleichwohl nutzen alle diese Systeme Kommunikationsformen, die deutliche Parallelen aufweisen: Sie basieren allesamt auf drahtloser Kommunikation über mehrere Hops; im Weiteren werden wir das breite Spektrum von Geräten und Anwendungen daher unter dem Begriff *drahtlose Multihop-Netzwerke* zusammenfassen.

Drahtlose Multihop-Netzwerke unterscheiden sich in einigen Punkten fundamental von anderen Netzwerkarchitekturen wie dem Internet und weisen viele einzigartige und interessante Eigenschaften auf. Die vielleicht bemerkenswerteste ist, dass beim Einsatz omnidirektionaler Antennen Übertragungen nicht nur vom angesprochenen Empfänger, sondern auch von anderen Netzwerkknoten in der Umgebung empfangen werden können. Darüber hinaus serialisiert das Medium Übertragungen lokal: Innerhalb einer beschränkten Umgebung können nicht mehrere Übertragungen gleichzeitig erfolgreich stattfinden. Wegen der sich daraus ergebenden Besonderheiten stellen drahtlose Multihop-Netzwerke existierende Kommunikationsprotokolle vor große Schwierigkeiten. Die Eigenschaften dieser Netze wurden daher bislang nahezu ausschließlich als Nachteil gesehen und auch als solcher behandelt.

Im Rahmen des Projekts *Kooperative schichtenübergreifende Überlastkontrolle für mobile Ad-Hoc-Netzwerke* werden die besonderen Eigenschaften eines drahtlosen Multihop-Netzwerkes aus einer anderen Perspektive betrachtet. Wie sich dabei herausstellt, können diese oft als Basis für neue, manchmal ungewöhnliche Lösungswege dienen. Statt existierende Ansätze so zu verändern, dass ihre schlechte Eignung für den Einsatz in drahtlosen Multihop-Umgebungen weniger deutlich in Erscheinung tritt, werden hier die Eigenschaften der Netzwerke für alternative Herangehensweisen gezielt ausgenutzt. Für eine Reihe von fundamentalen Problemen in drahtlosen Multihop-Netzwerken werden solche neuartigen Wege aufgezeigt. Dabei ist der Schlüsselgedanke stets derselbe: Die Eigenschaften des Übertragungsmediums lassen sich nutzen, um *implizit* Information zu gewinnen oder Vorgänge zu koordinieren, ohne dabei explizit Daten austauschen zu müssen.

Der Hauptfokus liegt hier auf der Funktionalität, die im Internet-Protokollstapel der Sicherungs- bzw. Medienzugriffsschicht und der Transportschicht zugeordnet ist. Alle vorgestellten Ansätze haben gemein, dass sie die Medieneigenschaften bereits in ihrem Ansatz berücksichtigen, und so in einem schichtenübergreifenden Ansatz auf einer gemeinsamen Grundlage zu Lösungen für verschiedenste Probleme führen. Zunächst werden die zentralen Funktionen des Transmission Control Protocol (TCP) aus neuen Blickwinkeln betrachtet. TCP realisiert im Internet-Protokollstapel Überlastkontrolle und zuverlässigen Ende-zu-Ende-Datentransport. Der im nachfolgenden Abschnitt vorgestellte Ansatz zur Überlastkontrolle namens CXCC resultiert aus diesen Überlegungen. CXCC wird durch den Zuverlässigkeitsmechanismus *BarRel* in Abschnitt 3. zu einer vollständigen TCP-Alternative ergänzt. Das in Abschnitt 4. diskutierte *BMCC* stellt eine Verallgemeinerung von CXCC auf den Fall der Multicast-Kommunikation dar. Anschließend wird in Abschnitt 5. aufgezeigt, wie durch die Ausnutzung der Medieneigenschaften und basierend auf einer Reinterpretation der zuvor eingeführten Ansätze die verfügbare Übertragungskapazität in bestimmten Situationen massiv gesteigert werden

kann. Dabei werden mehrere Paketübertragungen auf dem Medium im Inneren des Netzwerkes mittels Network Coding kombiniert. Nach der Diskussion dieser Protokolle für unterschiedliche Aufgaben in drahtlosen Multihop-Netzwerken werden in Abschnitt 6. Techniken zur Evaluation und Ergebnisinterpretation sowohl für simulative Untersuchungen als auch für Realwelt-Experimente vorgestellt, die im Rahmen der Entwicklung der genannten Protokolle entstanden sind und die für ihre Implementation und Untersuchung zum Einsatz kamen. Schließlich stellt Abschnitt 7. laufende Arbeiten in Richtung eines formalen Beweises der Korrektheit der Protokolle und Protokollimplementationen vor.

2. Implizite Überlastkontrolle: CXCC

Die erste Fragestellung, die hier betrachtet werden soll, ist die Überlastkontrolle. Ein Überlastkontrollmechanismus regelt die Datenrate, mit der ein Sender Daten in ein Netzwerk übertragen darf, um zu vermeiden dass die Rate der eingespeisten Daten die Kapazität des Netzwerkes übersteigt. Im Internet ist der Grund für den Verlust eines Paketes in der Regel ein Pufferüberlauf vor einer überlasteten Verbindung. Der Überlastkontrollmechanismus von TCP macht sich dies zunutze: TCP schließt aus dem Auftreten von Verlusten auf das Vorhandensein von Überlast und reduziert seine Senderate. Dies hat sich in drahtlosen Multihop-Netzwerken jedoch als problematisch erwiesen [5, 10]. Überlasteffekte äußern sich in drahtlosen Multihop-Netzwerken grundlegend anders. Wegen der lokal beschränkten Bandbreite sind in solchen Netzwerken nicht einzelne Verbindungen, sondern ganze (geografische) Bereiche des Netzwerkes überlastet. Außerdem treten häufig spontane Paketverluste auf, beispielsweise wegen Kollisionen oder aufgrund variierender Eigenschaften des Übertragungsmediums. Schon die Grundannahme der TCP-Überlastkontrolle, dass Paketverluste durch Pufferüberläufe entstehen und somit eine direkte Folge von Überlasterscheinungen sind, ist in drahtlosen Multihop-Netzwerken also falsch.

Deshalb ist es wichtig, alternative Möglichkeiten zur Überlastkontrolle zu betrachten. Die hier eingeführte *implizite schrittweise Überlastkontrolle* ist ein solcher Ansatz. Sie basiert auf dem Aufbau von Rückstau im Netzwerk mittels sehr kurzer Paketwarteschlangen. Die Grundidee ist dabei äußerst einfach: Die Übertragung eines Paketes an den Nachfolgeknoten entlang der Route zum Ziel wird verhindert, solange dieser Nachfolgeknoten nicht das vorangegangene Paket weitergeleitet hat. So wird ein Zufluss von Daten, der die verfügbare Transportkapazität übersteigt, vermieden. Umgesetzt und evaluiert wird dieses Konzept im Protokoll *Cooperative Cross-layer Congestion Control (CXCC)*, das in [15] vorgestellt wurde.

Die zentralen Faktoren, die einen solchen Ansatz in drahtlosen Multihop-Netzwerken nicht nur ermöglichen, sondern ihn sogar zu einer besonders guten Wahl

machen, sind die speziellen Eigenschaften des Mediums. Um dies zu verstehen, ist es hilfreich, das konkrete Vorgehen des Protokolls beim Weiterleiten von Daten einer Verbindung zu betrachten. Schematisch ist es in Abbildung 1 dargestellt, die den Ablauf einer Datenübertragung von einer Quelle links über zwei Zwischenstationen zu einem Zielknoten rechts zeigt.

Nach dem Weiterleiten eines Paketes muss ein Knoten warten (er ist „blockiert“), bis er erfährt, dass sein Nachfolger das Paket abermals weitergeleitet hat. Er kann dies erfahren, indem er das Weiterleiten seines Nachfolgers mithört. Dieses Mithören wird durch das Übertragungsmedium ermöglicht. Es liefert einerseits ein *implizites Acknowledgment*, also eine indirekte Bestätigung, dass die vorherige Übertragung zum Nachfolger erfolgreich war, ohne dass ein separates Bestätigungspaket nötig wäre. Gleichzeitig löst es die Blockierung und erlaubt die Übertragung eines Folgepaketes. Nur der endgültige Zielknoten leitet das Paket nicht weiter und muss deshalb seinem unmittelbaren Vorgänger den Empfang explizit bestätigen.

Noch zentraler tritt die Berücksichtigung der Medieneigenschaften in einem anderen Zusammenhang hervor: Während der Übertragung eines Folgepaketes in einen Knoten hinein kann das Medium nicht gleichzeitig dafür genutzt werden, ein früher eingetroffenes Paket aus diesem Knoten heraus weiterzuleiten. Im Gegensatz zu kabelgebundenen Netzen ist es in drahtlosen Multihop-Umgebungen deshalb *nicht* sinnvoll, mehr als ein Paket in jeder Station entlang der Route zwischenspeichern. CXCC berücksichtigt diese besondere Eigenschaft des Mediums: Der beschriebene Mechanismus gibt einem Knoten nach dem Erhalt eines Paketes stets Gelegenheit, dieses auch weiterzuleiten, bevor sein Vorgänger ihm ein Folgepaket zusenden darf. Dies hat auch zur Folge, dass jeder Zwischenknoten maximal ein Paket puffert.

Durch den einfachen Mechanismus des Blockierens baut sich ein Rückstau in Richtung Quelle auf, wenn die Übertragung an einer Stelle im Netz verzögert ist. Letztlich kann damit auch der Quellknoten Pakete nicht schneller ins Netzwerk einspeisen, als diese es passieren können – das Resultat ist also ein Überlastkontrollmechanismus. Das Besondere dieser Herangehensweise ist, dass die Überlastkontrolle rein implizit erfolgt: Es gibt keinen dedizierten Mechanismus oder Algorithmus zur Festlegung der Senderate, kein Knoten muss explizit Übertragungsraten schätzen und keine Komponente des Netzwerks muss Rückmeldung über den aktuellen Lastzustand geben. Die einfache Regel des Abwartens nach der Übertragung eines Paketes bis zum Mithören des Weiterleitens bringt ein selbstregulierendes System hervor.

Selbstverständlich ist von diesem Ansatz bis zu einem vollständigen Protokoll noch ein weiter Weg zu gehen. Insbesondere muss die Möglichkeit von (bei drahtlosen Übertragungen durchaus häufigen) Übertragungsfehlern oder nicht mehr er-

reichbaren Nachfolgeknoten (beispielsweise in mobilen Netzen aufgrund der Bewegung der Knoten) in Betracht gezogen werden. Natürlich darf ein Knoten nach einer fehlgeschlagenen Übertragung nicht unbegrenzt darauf warten, dass sein Nachfolger ein Paket weiterleitet, das dieser nie erhalten hat.

Um mit beiden Problemen effektiv und effizient umzugehen kommt bei CXCC ein Mechanismus namens *Request for Acknowledgment* (RFA) zum Einsatz. Mit einem RFA-Paket kann ein Knoten nach Versand eines Paketes seinen Nachfolger entlang der Route fragen, ob dieser eine vorangegangene Übertragung erhalten hatte. Beispielhaft ist das in Abbildung 2 zu sehen, wo der Fall ein verlorengangenes implizites Acknowledgment durch einen RFA-ACK-Handshake aufgelöst wird. Trotz – oder gerade wegen – der Vielfalt von Möglichkeiten, wie Verluste und Verzögerungen beim Weiterleiten in einer so komplexen und unberechenbaren Umgebung wie einem drahtlosen Multihop-Netzwerk zusammenspielen können, zeigt ein RFA-basierter Ansatz im Vergleich zur gängigen Lösung unmittelbarer Neuübertragungen beim zu langen Ausbleiben von (impliziten) Bestätigungen eine deutlich bessere Performance im Zusammenspiel mit impliziter schrittweiser Überlastkontrolle. Abbildung 3 demonstriert die Wirksamkeit der CXCC-Überlastkontrolle in einem sehr einfachen ns-2-Simulationsszenario mit einer 10-Hop-Kettentopologie und bidirektionalem UDP-Datenverkehr. Wie zu erwarten bricht der erreichte Durchsatz bei Verwendung von IEEE 802.11 ohne Überlastkontrollmechanismus rapide ein, sobald ein kritischer Wert der durch die Anwendungen in den beiden Endknoten angebotenen Last überschritten wird. Mit Überlastkontrolle mittels CXCC wird hingegen – bei Verwendung von RFA – die Quelldatenrate durch Rückstau zu den Quellknoten so geregelt, dass der maximal mögliche Durchsatz unabhängig von der angebotenen Last realisiert werden kann. In weitergehenden Vergleichsuntersuchungen zeigt sich eine klare Überlegenheit gegenüber TCP-basierten Ansätzen zur Überlastkontrolle in drahtlosen Multihop-Umgebungen sowohl in Hinblick auf den erreichten Durchsatz als auch bezüglich anderer wichtiger Eigenschaften wie Übertragungsverzögerungen und Protokoll-Overhead. Eine tiefergehende Diskussion der Ergebnisse und ihrer Zusammenhänge sowie Details zu CXCC finden sich in [15].

3. Implizite Zuverlässigkeit: BarRel

Einer der Gründe für die überragenden Ergebnisse von CXCC im Vergleich zu anderen Ansätzen ist die Tatsache, dass für die Überlastkontrolle keine Kontrollpakete notwendig sind, die wie die Bestätigungspakete von TCP vom Ziel zurück zur Quelle laufen. Solche Pakete wirken sich in drahtlosen Multihop-Netzwerken sehr nachteilig aus [4], unter anderem weil sie mit den Datenpaketen um das Medium konkurrieren und häufig die Ursache von Kollisionen und damit Paket-

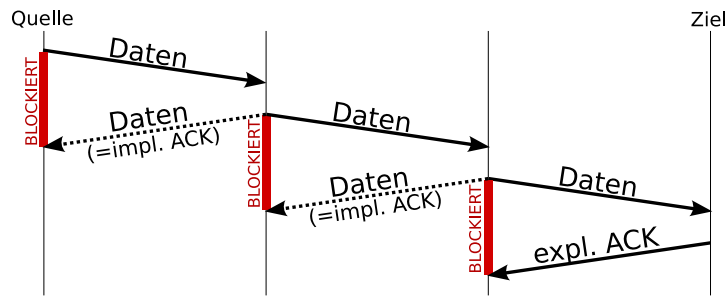


Abbildung 1. Paketweiterleitung in CXCC.

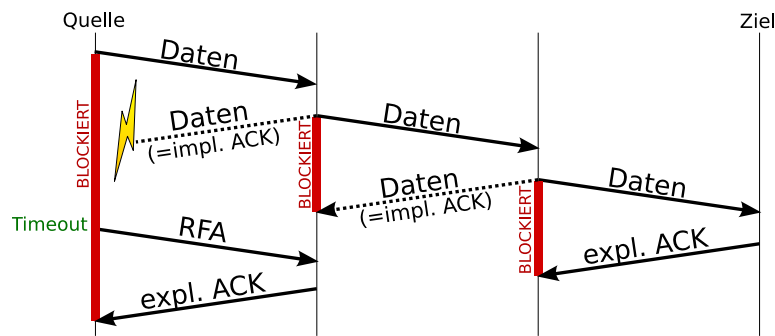


Abbildung 2. Erholung vom Verlust eines impliziten Acknowledgment mittels RFA.

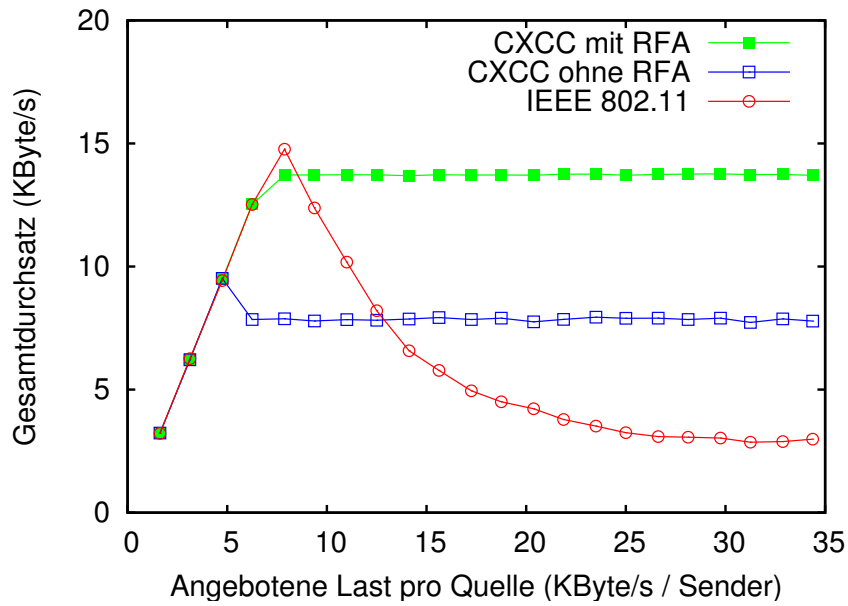


Abbildung 3. Durchsatz mit CXCC in bidirektionaler Kettentopologie.

verlustrufen sind. Allerdings vermeidet CXCC zwar Ende-zu-Ende-Kontrolldatenverkehr für die Überlastkontrolle, gewährleistet dabei aber keine TCP-äquivalente Ende-zu-Ende-Zuverlässigkeit. Hierfür könnten wiederum explizite Ende-zu-Ende-Bestätigungspakete notwendig sein. In der Tat wurde bislang allgemein angenommen, dass sie für eine zuverlässige Ende-zu-Ende-Datenübertragung unvermeidlich sind.

Mit dem Protokoll *Backpressure Reliability* (BarRel), dessen Details in [11] diskutiert werden, konnte konstruktiv gezeigt werden, dass dies nicht zutrifft. BarRel verfolgt den mit CXCC eingeschlagenen Weg weiter und ergänzt die implizite Überlastkontrolle durch einen Mechanismus für die zuverlässige Ende-zu-Ende-Übertragung von Daten. Es nutzt Wissen über die Funktionsweise der CXCC-Überlastkontrolle, um implizit auf die erfolgreiche Zustellung von Datenpaketen an einen weiter entfernten Zielknoten zu schließen. Im Gegensatz zu existierenden TCP-äquivalenten Transportprotokollen benötigt BarRel keinen kontinuierlichen Strom von Bestätigungspaketen und vermeidet dadurch den problematischen gegenläufigen Kontrolldatenverkehr.

Die Idee, die BarRel zugrunde liegt, nutzt CXCCs Limitierung der Zahl von zwischengespeicherten Paketen in den Knoten entlang der Route. Wie zuvor gesehen, beschränkt CXCC die Länge der Paketwarteschlange in jedem Zwischenknoten auf ein Paket. Wir nehmen nun an, dass die Routenlänge n beim Quellknoten bekannt ist; dies ist mit vielen Routingansätzen problemlos realisierbar. Verschickt der Quellknoten dann das $i + n$ -te Paket, so kann er daraus schließen, dass das i -te Paket beim Zielknoten eingetroffen sein muss – andernfalls hätte CXCC den Versand dieses Paketes nicht erlaubt!

Auch hier stellen sich wieder eine ganze Reihe von praktischen Herausforderungen, wenn die Idee in einem lauffähigen Protokoll umgesetzt werden soll. So stellt sich etwa die Frage, wie fehlgeschlagene Paketübertragungen und Änderungen der Route zum Ziel behandelt werden sollen. Es zeigt sich jedoch, dass diese Probleme lösbar sind und dass es möglich ist, TCP vollständig und transparent durch CXCC und BarRel zu ersetzen. Die wahrscheinlich interessanteste Frage im Zusammenhang mit bestätigungsfreier Ende-zu-Ende-Zuverlässigkeit ist die nach dem Verhalten beim (vorübergehenden oder endgültigen) Ende einer Datenübertragung. Denn wenn der Versand des $i + n$ -ten Paketes dazu dient, indirekt den Erhalt des i -ten Paketes zu bestätigen, dann muss selbstverständlich auch der Fall berücksichtigt werden, dass es kein $i + n$ -tes Paket gibt.

Der erste und offensichtliche Ansatz ist der Versand eines einzelnen Bestätigungspaketes vom Ziel zurück zur Quelle für das letzte Datenpaket. Dies löst das Problem, widerspricht aber dem eigentlich rein impliziten Ansatz. Eine überraschend einfache, pragmatische Lösung entschärft das Problem des fehlenden $i + n$ -ten Paketes auf andere Weise: Wenn die Anwendung keine weiteren Daten

mehr liefert, die durch ihren Versand für die implizite Bestätigung früher versandter Pakete dienen können, so kann das Protokoll schlicht den Sendepuffer mit n leeren Paketen – in BarRel *Capacity-Refill-Pakete* (CaRe-Pakete) genannt – auffüllen. Wurde das letzte CaRe-Paket verschickt, so muss zuvor das letzte „wichtige“ Datenpaket den Zielknoten erreicht haben. Damit wird zuverlässige Datenübertragung ohne jeglichen Kontrollpaketfluss in Gegenrichtung möglich.

Der Versand von n zusätzlichen CaRe-Paketen am Übertragungsende wirkt auf den ersten Blick ineffizient, weist aber bei genauerer Betrachtung durchaus Vorteile auf. Insbesondere ist es bei der Verwendung von CaRe-Paketen nicht notwendig, eine Zeit festzulegen, die maximal auf ein Bestätigungspaket gewartet werden soll, bevor ein Paketverlust angenommen wird. Die Festlegung dieser Wartezeit ist in zuverlässigen Datenübertragungsprotokollen stets ein großes Problem, und wurde in der Vergangenheit intensiv erforscht. Dennoch bleibt sie ein potentieller wunder Punkt eines jeden Protokollentwurfs, der Bestätigungspakete einsetzt – einschließlich TCP und der BarRel-Variante mit einem einzelnen Bestätigungspaket am Übertragungsende. Bei Verwendung von CaRe-Paketen ist die Wahl einer solchen Wartezeit nicht nötig, wodurch das Problem vollständig vermieden wird.

4. Rückdruckbasierte Multicast-Überlastkontrolle: BMCC

Die implizite schrittweise Überlastkontrolle kann auch auf Multicast-Datenübertragungen angewendet werden, also auf Kommunikationsszenarien, in denen ein Sender identische Daten zeitgleich an mehrere Empfänger versenden möchte. Dies wird im Protokoll *Backpressure Multicast Congestion Control* (BMCC) umgesetzt, dessen Details in [16] beschrieben werden. BMCC erzielt eine effektive Regelung der Quelldatenrate bei geringen Paketlaufzeiten und minimalem Kontrolldatenaufkommen.

Für die Zustellung an eine Gruppe von Empfängern statt eines einzelnen Zielknotens werden die Daten entlang einer Baumstruktur weitergeleitet, deren Wurzel der Quellknoten ist und deren Blätter von Zielknoten gebildet werden. Ein weiterleitender Knoten muss die Daten also an mehr als einen Nachfolger weiterreichen, wenn sich bei ihm der Multicast-Baum verzweigt. Wird das Rückdruckprinzip der impliziten schrittweisen Überlastkontrolle auf diese Situation erweitert, so müssen zunächst die Mechanismen für implizite und explizite Bestätigungen, zur Erkennung und Behebung von Übertragungsfehlern und zum Identifizieren von nicht mehr erreichbaren Nachfolgeknoten entsprechend verallgemeinert werden. Im konkreten Fall wurde dies in Zusammenarbeit mit der Universität Mannheim für das dort entwickelte geographische Multicast-Routingprotokoll Scalable Position-Based Multicast (SPBM) [17] implementiert und untersucht. Da bei geographischen Ansätzen zum Multicast-Routing dem Quellknoten die einzelnen Gruppenmitglie-

der ebenso wenig bekannt sind wie deren genaue Anzahl, ist das Durchführen von Überlastkontrolle hier besonders anspruchsvoll.

Eine naheliegende Erweiterung des CXCC-Rückdruckmechanismus erlaubt einem Knoten die Übertragung eines Folgepaketes, sobald *alle* seine Nachfolger das vorangegangene Paket weitergeleitet haben. Wenn dies jeder Knoten im Multicast-Baum so handhabt, dann wird sich Rückdruck entsprechend der „engsten“ Stelle im gesamten Baum aufbauen. Die Senderate der Quelle wird sich also nach dem langsamsten aller Empfänger richten. Dies kann durchaus erwünscht sein, wenn es wichtig ist, allen Empfängern alle Daten (oder zumindest einen Großteil) zu übermitteln. Ist jedoch auch der Empfang eines mehr oder weniger großen Bruchteils der Daten für einen Empfänger nützlich – etwa, weil entsprechende Codierungsverfahren zum Einsatz kommen –, dann bleibt gegebenenfalls viel Medienkapazität auf dem Weg zu besser erreichbaren Empfängern ungenutzt. Eine alternative Version von BMCC erlaubt deshalb, basierend auf einer Relaxation des Rückstauprinzips, die Zustellung mit höheren Datenraten an besser erreichbare Empfänger, ohne dass hierfür die langsamen Empfängern zur Verfügung stehende Medienkapazität beeinträchtigt wird.

5. Koordiniertes Network Coding: noCoCo

Network Coding kombiniert – in der Regel mittels Linearkombinationen – im Inneren eines Netzwerkes mehrere Pakete in eine einzige Übertragung. Ursprünglich wurde dies in [2] vorgeschlagen; viele theoretische Arbeiten zeigen, dass dadurch große Kapazitätsgewinne möglich sind. Ein einfaches Beispiel in drahtlosen Netzwerken zeigt Abbildung 4. In Abbildung 4(a) werden zwei Pakete übertragen, Paket *A* vom linken zum rechten und Paket *B* vom rechten zum linken Knoten. Beide werden vom mittleren Netzwerkknoten weitergeleitet. Insgesamt sind somit vier Übertragungen notwendig. In Abbildung 4(b) werden die gleichen Pakete ausgetauscht. Statt jedoch *A* und *B* einzeln weiterzuleiten, überträgt der Knoten in der Mitte das bitweise Exklusiv-Oder $A \oplus B$. Da jeder der Endknoten das von ihm selbst erzeugte Paket kennt, können beide die Operation umkehren und die für sie bestimmten Pakete zurückgewinnen. Dadurch wird eine der vier Übertragungen eingespart. Dies lässt sich auf komplexere Situationen verallgemeinern.

Von Katti et al. wurde Network Coding im Protokoll COPE [8] erstmals praktisch in drahtlosen Multihop-Netzwerken angewendet. Dort geschieht dies rein *opportunistisch*: Kombinierbare Übertragungen werden zusammengefasst, falls sich Gelegenheiten hierfür spontan ergeben. Dies nutzt die Möglichkeiten jedoch nicht vollständig aus. Gerade in kritischen Situationen stellen sich Gelegenheiten zum Network Coding nur vergleichsweise selten spontan ein. Der in Kooperation mit der Universität Cambridge entwickelte Ansatz *Near-Optimal Co-ordinated Coding*

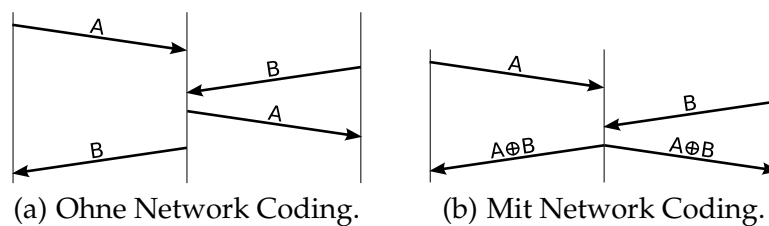


Abbildung 4. Paketaustausch mit und ohne Network Coding.

(noCoCo) [14] ersetzt das rein opportunistische Vorgehen von COPE in Situationen, in denen gegenläufiger Datenverkehr dieselben Zwischenstationen passiert. Es stellt sich heraus, dass dies ideal mit der impliziten schrittweisen Überlastkontrolle in CXCC kombiniert werden kann. CXCC wird deshalb als Basis benutzt und die zugrunde liegende Idee der impliziten Koordination wird weiterentwickelt, um das Zusammenspiel von Übertragungen benachbarter Netzwerkknoten zu steuern. So wird es mit noCoCo möglich, Gelegenheiten zum Network Coding nicht nur spontan zu nutzen, sondern ihre Existenz für bidirektionalen Datenverkehr zu *garantieren*. Dadurch kann der höchstmögliche Gewinn tatsächlich praktisch realisiert werden.

Analytisch lassen sich Zusammenhänge zwischen der Zahl im Netzwerk gepufferter Pakete und dem mit Network Coding in bidirektionalem Datenverkehr realisierbaren Gewinn herleiten; es zeigt sich, dass noCoCo dem theoretischen Optimum extrem nahe kommt. Vergleichende Untersuchungen mit COPE zeigen, dass sich der Datendurchsatz mit koordiniertem Network Coding oft dramatisch steigern lässt, und dass gleichzeitig die Paketlaufzeit und der durch Kontrolldaten und Übertragungswiederholungen verursachte Overhead stark vermindert werden. Abbildung 5 zeigt im gleichen einfachen Simulationsszenario wie zuvor in der Diskussion von CXCC, dass sich durch opportunistisches Network Coding mit COPE der Durchsatz (gleichgültig ob mit oder ohne Einsatz von CXCC) zwar geringfügig steigern lässt, die Vorteile jedoch dramatisch höher sind, wenn Network Coding mit noCoCo koordiniert wird. Abbildung 6 analysiert dasselbe Szenario aus einem anderen Blickwinkel und betrachtet die Zahl an Bytes, die durchschnittlich über das drahtlose Medium übertragen werden, um ein Byte Nutzlast einen Hop näher an sein Ziel zu bringen – einschließlich aller Kontrollnachrichten, Header, Übertragungswiederholungen etc.. Besonders erstaunlich ist hierbei, dass aufgrund der hohen Gewinne durch das koordinierte Network Coding mit noCoCo im Mittel *weniger* als ein Byte auf dem Medium übertragen werden muss, um ein Byte Nutzlast weiterzutransportieren. Eine tiefere Diskussion der Ergebnisse findet sich in [14].

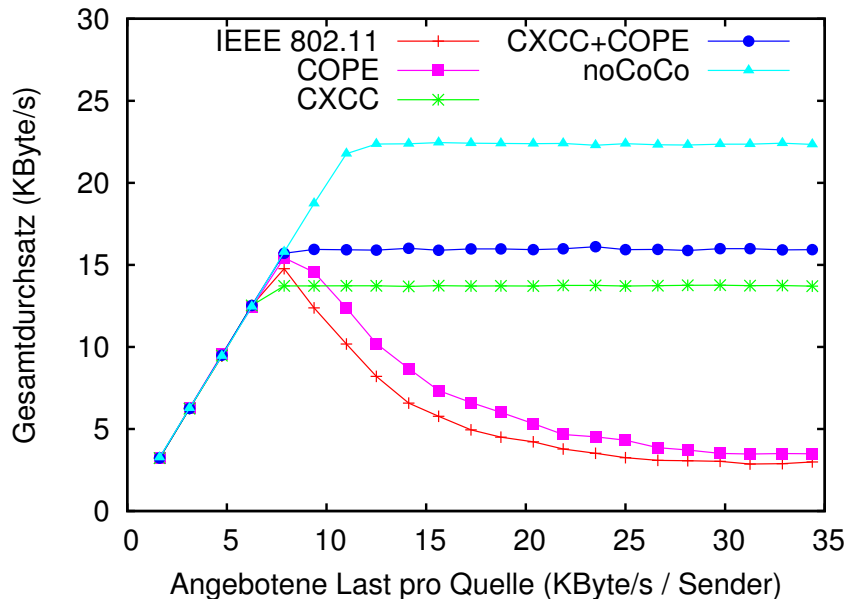


Abbildung 5. Durchsatz mit noCoCo in bidirektionaler Kettentopologie.

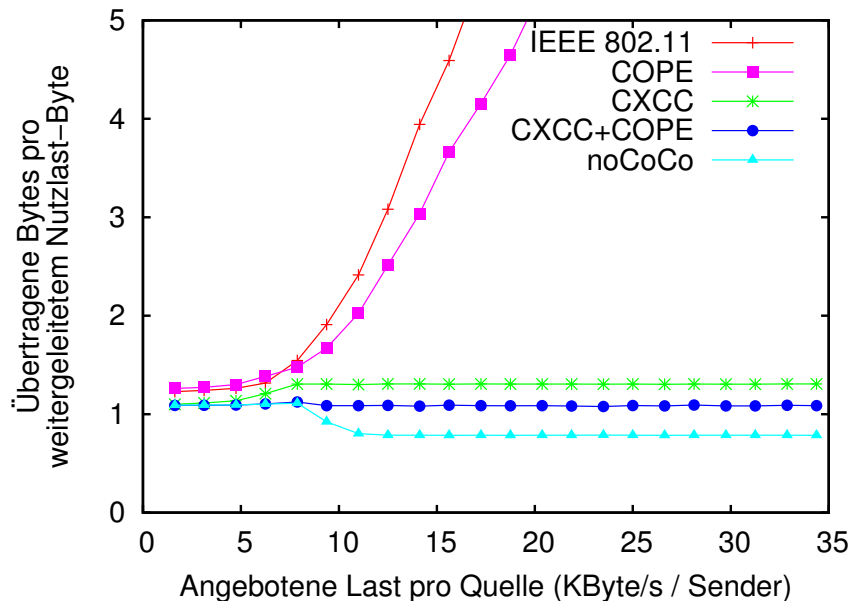


Abbildung 6. Protokolloverhead mit noCoCo in bidirektionaler Kettentopologie.

6. Evaluation schichtübergreifender Protokolle

Die Untersuchung der vorgestellten Protokolle erfolgte einerseits durch ausführliche Simulationsstudien, darüber hinaus jedoch auch durch Realwelt-Messungen auf Basis mehrerer unabhängiger Implementationen. Bereits im Falle der Untersuchungen mit Netzwerksimulatoren und noch deutlicher bei den realen Implementationen ergeben sich dabei praktische Schwierigkeiten aus der schichtübergreifenden Architektur der Protokolle. Die an mehreren Stellen notwendig werdenden Eingriffe sind eine Herausforderung bei der Implementation in einem Netzwerksimulator ebenso wie beim Einbinden in die Protokollarchitektur realer Systeme. Bei der Auswertung von Simulationen und Messungen wird erneut deutlich, dass die existierenden Werkzeuge und Vorgehensweisen die Untersuchung von schichtübergreifenden Ansätzen nur eingeschränkt unterstützen.

Im Falle der Simulationen ist das Verfolgen von Vorgängen über mehrere Schichten und Knoten hinweg essentiell wichtig für das Verständnis von Protokollverhalten und -fehlverhalten. Genau dieses Verfolgen von Ereignisabfolgen ist jedoch mit den üblicherweise bereitgestellten Analysemöglichkeiten unnötig schwierig. In Kooperation mit der Universität Mannheim wurde deshalb ein Werkzeug entwickelt, das die interaktive Analyse von Simulationsläufen des Netzwerksimulators ns-2 auch in komplexen Netzwerkumgebungen unterstützt. Das entwickelte Werkzeug namens Huginn [12, 13] bietet eine Vielzahl von verschiedenen Darstellungsmöglichkeiten über in die Visualisierung des Netzwerkgeschehens eingebettete dynamisch berechnete Auswertungen. Es erlaubt dadurch dem Entwickler eines Protokolls, das Geschehen im Netzwerk interaktiv zu erkunden und bis hinab zu einzelnen Übertragungsvorgängen zu analysieren.

Um Untersuchungen von Realwelt-Implementationen schichtübergreifender Protokolle bis hinab zur Medienzugriffsschicht zu ermöglichen ist zunächst eine Testplattform notwendig, in der sich die entsprechenden Bestandteile des Protokollstapels überhaupt modifizieren lassen. Auf Basis gängiger WLAN-Hardware sind Modifikationen, wie sie beispielsweise für das Protokoll CXCC notwendig sind, nicht ohne weiteres möglich. Für eine Implementation von CXCC und entsprechende Messungen kamen daher die ESB-Sensorknoten der Freien Universität Berlin zum Einsatz. Um die Durchführung von Experimenten auf diesen Geräten und die Auswertung der Resultate zu ermöglichen, wurden ein umfassender Satz von Protokollen, Mechanismen und Werkzeugen entwickelt, die wir in [6, 7] näher vorstellen. Beispielsweise unterstützt das dort beschriebene ESB-basierte Cross-Layer-Testbett die Analyse und gezielte Beeinflussung von Netzwerktopologie und Routing-Tabellen sowie die speicherplatz- und rechenzeiteffiziente Aufzeichnung von Logdaten während der Durchführung von Messläufen.

7. Formale Verifikation der Protokolle

Die hier bislang diskutierten Protokolle für drahtlose Multihop-Netzwerke haben ihre Funktion und Performance in Simulationen und teilweise auch in realen Implementationen unter Beweis gestellt. Die Korrektheit der Protokollspezifikation einerseits und der konkreten Implementationen andererseits lassen sich jedoch mit solchen Untersuchungen alleine niemals zweifelsfrei belegen. Insbesondere in Anwendungsfällen, die entweder inhärent sicherheitskritisch sind (etwa im Bereich Fahrzeug-zu-Fahrzeug-Kommunikation oder beim Einsatz von drahtlosen Multihop-Netzwerken in Katastrophenszenarien) oder in denen Modifikationen am einmal ausgebrachten System schwierig und aufwändig sind (etwa in Sensornetzen mit weit verstreuten, schwer zugänglichen Netzwerkknoten) wäre jedoch eine solche zweifelsfreie Untermauerung der Korrektheit höchst wünschenswert.

Es existieren eine ganze Reihe von Methoden zur formalen Verifikation von Softwaresystemen, die prinzipiell auch für Netzwerkprotokolle zum Einsatz kommen können. Es stellt sich jedoch bei genauerer Betrachtung heraus, dass gerade drahtlose Multihop-Netzwerke aufgrund ihres hohen Grades an Parallelität und der vielen nichtdeterministischen Elemente in ihrem Verhalten solche Ansätze vor große Herausforderungen stellen. Die formale Modellierung von Protokolle in drahtlosen Multihop-Umgebungen wurde noch nicht in signifikanter Tiefe wissenschaftlich untersucht, entsprechende Ansätze befinden sich noch in einem sehr frühen Stadium.

Um Erfahrungen mit der Anwendung formaler Methoden für drahtlose Multihop-Kommunikationsprotokolle zu sammeln und dabei gleichzeitig weitere Erkenntnisse über das Verhalten der entworfenen Protokolle zu gewinnen wurden deshalb in Kooperation mit der Arbeitsgruppe für Softwaretechnik der Universität Düsseldorf erste Schritte hin zu einem formalen Modell der Protokolle CXCC und BarRel unternommen. Da die Grundstruktur dieser Protokolle trotz ihrer großen Effizienz auf wenigen, einfachen Prinzipien und Regeln beruht, erscheinen sie als besonders gut geeignet, um trotz vollständiger Abbildung der Funktionalität die Größe und Komplexität der Modelle beherrschbar zu halten.

Einerseits wird dabei auf einen Beweis dafür hingearbeitet, dass die Kombination aus CXCC und BarRel in ihrer Funktionalität äquivalent zu TCP ist, andererseits zeichnet sich daran anschließend die Möglichkeit der Herleitung einer beweisbar korrekten Protokollimplementation ab. Für die entsprechenden Arbeiten kommen Event-B [1] als formale Modellierungsmethode und das Werkzeug ProB [9] zum Einsatz. Die entsprechenden Arbeiten dauern noch nach, erste Zwischenergebnisse wurden in [3] vorgestellt.

8. Zusammenfassung

Die vorgestellten Protokolle und Lösungsansätze für verschiedenartige Probleme in drahtlosen Multihop-Netzwerken haben in ihrem Kern alle gemeinsam, dass sie die Eigenschaften des zugrundeliegenden Systems aus einem neuen Blickwinkel betrachten. Statt des Versuchs, die von anderen Netzen stark abweichenden Faktoren durch Modifikationen zu kompensieren oder zu kaschieren, werden die Besonderheiten der spezifischen Umgebung als etwas Positives aufgefasst, das selbst zu effektiven Lösungen beitragen kann.

Vorgeschlagen wurden Protokolle und Mechanismen zur Überlastkontrolle mittels impliziten Rückdrucks in CXCC, BarRel für die zuverlässige, TCP-äquivalente Ende-zu-Ende-Datenübertragung ohne kontinuierlichen Strom von Ende-zu-Ende-Acknowledgments, eine Verallgemeinerung der CXCC-Überlastkontrolle für Multicast-Datenverkehr namens BMCC und die Koordination lokaler Network-Coding-Operationen auf Basis der CXCC-Grundprinzipien in noCoCo. Jeder der vorgeschlagenen Ansätze nutzt Wissen über die Eigenschaften des Netzwerks, um Informationen implizit zu gewinnen, und alle zeigen, dass dadurch neue und manchmal unkonventionelle, doch höchst effektive Lösungen möglich werden. Darüber hinaus wurden Werkzeuge und Vorgehensweisen entwickelt, die die Untersuchung von schichtübergreifenden Protokollen sowohl simulativ als auch mit realen Experimenten signifikant erleichtern oder gar erst ermöglichen. Schließlich zeigen erste Ergebnisse aus der Modellierung und Untersuchung der vorgeschlagenen Protokolle mittels formaler Methoden, dass durch die gezielte Weiterentwicklung und den konsequenten Einsatz entsprechender Techniken beweisbar korrektes Verhalten auch in komplexen Umgebungen wie drahtlosen Multihop-Netzwerken durchaus erreichbar sein kann.

Literatur

- [1] ABRIAL, JEAN-RAYMOND, DOMINIQUE CANSSELL und DOMINIQUE MÉRY: *Refinement and Reachability in Event \mathcal{P}* . In: *ZB '05: Proceedings of the 4th International Conference of Z and B Users*, Seiten 222–241, April 2005.
- [2] AHLWEDE, RUDOLF, NING CAI, SHUO-YEN LI und RAYMOND YEUNG: *Network Information Flow*. *IEEE Transactions on Information Theory*, 46(4):1204–1216, 2000.
- [3] BENDISPOSTO, JENS, MICHAEL JASTRAM, MICHAEL LEUSCHEL, CHRISTIAN LOCHERT, BJÖRN SCHEUERMANN und INGO WEIGELT: *Validating Wireless Congestion Control and Reliability Protocols using ProB and Rodin*. In: *FMWS '08: Proceedings of the International Workshop on Formal Methods for Wireless Systems*, August 2008.
- [4] DE OLIVEIRA, RUY und TORSTEN BRAUN: *A Smart TCP Acknowledgment Approach for Multihop Wireless Networks*. *IEEE Transactions on Mobile Computing*, 6(2):192–205, Februar 2007.

- [5] FU, ZHENGHUA, XIAOQIAO MENG und SONGWU LU: *How Bad TCP Can Perform In Mobile Ad Hoc Networks*. In: *ISCC '02: Proceedings of the 7th IEEE International Symposium on Computers and Communication*, Seiten 298–303, Juli 2002.
- [6] JERSCHOW, YVES IGOR, BJÖRN SCHEUERMANN, CHRISTIAN LOCHERT und MARTIN MAUVE: *A Cross-Layer Protocol Evaluation Framework on ESB Nodes (Demo)*. In: *REALMAN '06: Proceedings of the 2nd International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality*, Seiten 104–106, Mai 2006.
- [7] JERSCHOW, YVES IGOR, BJÖRN SCHEUERMANN, CHRISTIAN LOCHERT und MARTIN MAUVE: *A Real-World Framework to Evaluate Cross-Layer Protocols for Wireless Multihop Networks*. In: *REALMAN '06: Proceedings of the 2nd International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality*, Seiten 1–6, Mai 2006.
- [8] KATTI, SACHIN, HARIHARAN RAHUL, WENJUN HU, DINA KATABI, MURIEL MEDARD und JON CROWCROFT: *XORs in The Air: Practical Wireless Network Coding*. In: *SIGCOMM '06: Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Seiten 243–254, September 2006.
- [9] LEUSCHEL, MICHAEL und MICHAEL BUTLER: *ProB: A Model Checker for B*. In: *FM '03: Proceedings of the 12th International FME Symposium*, Seiten 855–874, September 2003.
- [10] RAGHUNATHAN, VIVEK und PANGANAMALA R. KUMAR: *A Counterexample in Congestion Control of Wireless Networks*. *Elsevier Performance Evaluation*, 64:399–418, Juni 2006.
- [11] SCHEUERMANN, BJÖRN: *Reading Between the Packets – Implicit Feedback in Wireless Multihop Networks*. Doktorarbeit, Heinrich-Heine-Universität, Düsseldorf, Oktober 2007.
- [12] SCHEUERMANN, BJÖRN, HOLGER FÜSSLER, MATTHIAS TRANSIER, MARCEL BUSSE, MARTIN MAUVE und WOLFGANG EFFELSBERG: *Huginn: A 3D Visualizer for Wireless ns-2 Traces*. In: *MSWiM '05: Proceedings of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Seiten 143–150, Oktober 2005.
- [13] SCHEUERMANN, BJÖRN, HOLGER FÜSSLER, MATTHIAS TRANSIER, MARTIN MAUVE und WOLFGANG EFFELSBERG: *Visualizing Wireless ns-2 Traces in 3D*. In: *MobiCom '05: The 11th Annual ACM International Conference on Mobile Computing and Networking, Demo Session*, September 2005.
- [14] SCHEUERMANN, BJÖRN, WENJUN HU und JON CROWCROFT: *Near-Optimal Coordinated Coding in Wireless Multihop Networks*. In: *CoNEXT '07: Proceedings of the 3rd International Conference on Emerging Networking Experiments and Technologies*, Dezember 2007.
- [15] SCHEUERMANN, BJÖRN, CHRISTIAN LOCHERT und MARTIN MAUVE: *Implicit Hop-by-Hop Congestion Control in Wireless Multihop Networks*. *Elsevier Ad Hoc Networks*, 6(2):260–286, April 2008.
- [16] SCHEUERMANN, BJÖRN, MATTHIAS TRANSIER, CHRISTIAN LOCHERT, MARTIN MAUVE und WOLFGANG EFFELSBERG: *Backpressure Multicast Congestion Control in Mobile Ad-Hoc Networks*. In: *CoNEXT '07: Proceedings of the 3rd International Conference on Emerging Networking Experiments and Technologies*, Dezember 2007.
- [17] TRANSIER, MATTHIAS, HOLGER FÜSSLER, JÖRG WIDMER, MARTIN MAUVE und WOLFGANG EFFELSBERG: *A Hierarchical Approach to Position-Based Multicast for Mobile Ad-hoc Networks*. *Springer Wireless Networks*, 13(4):447–460, August 2007.