

Wenn sich Rechner blind verstehen: Implizites Feedback in drahtlosen Multihop-Netzwerken

Björn Scheuermann

Heinrich-Heine-Universität Düsseldorf
scheuermann@cs.uni-duesseldorf.de

Abstract: Beim Einsatz gängiger und bewährter Netzwerkprotokolle in drahtlosen Multihop-Netzwerken treten große Schwierigkeiten zu Tage. Diese resultieren aus den besonderen Eigenschaften solcher Netzwerke. Deshalb wurden diese Eigenschaften bislang nahezu ausschließlich negativ gesehen, und viele Versuche sind unternommen worden, existierende Protokolle so anzupassen, dass sie die auftretenden Effekte besser tolerieren können. Der hier verfolgte Ansatz unterscheidet sich hiervon grundlegend. Die Eigenschaften drahtloser Multihop-Netzwerke werden hier aus einem anderen Blickwinkel betrachtet. Es werden Mechanismen vorgestellt, die sie gezielt nutzen, um verschiedene Probleme in diesen Netzwerken auf neuartige Weise anzugehen. Dabei zeigt sich, dass sich gerade die Besonderheiten der Umgebung als Grundlage für alternative, oft unkonventionelle Lösungsansätze eignen.

1 Einführung

In drahtlosen Multihop-Netzwerken können die Teilnehmer auch ohne fest installierte Infrastruktur über ihre eigene Funkreichweite hinaus kommunizieren, indem die verschickten Daten über mehrere drahtlose Zwischenschritte von Netzwerkteilnehmer zu Netzwerkteilnehmer in Richtung ihres Zieles weitergereicht werden. Der Einsatz solcher Netze wird für eine Reihe von Anwendungen diskutiert, darunter beispielsweise die Vernetzung von Fahrzeugen auf der Straße, um die Verkehrssicherheit und den Fahrkomfort erhöhen, Mesh-Netzwerke, die die Reichweite von Internetzugangspunkten vergrößern, oder die Kommunikation in Katastrophenfällen, wenn keine Infrastruktur verfügbar ist.

Drahtlose Multihop-Netzwerke unterscheiden sich in einigen Punkten fundamental von gängigen 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. Daher wurden die Eigenschaften dieser Netze bislang nahezu ausschließlich als Nachteil gesehen und auch als solcher behandelt.

Hier werden die besonderen Eigenschaften eines drahtlosen Multihop-Netzwerkes aus einer anderen Perspektive betrachtet. Wie sich dabei herausstellt, können sie oft als Basis für neue, manchmal ungewöhnliche Lösungswege dienen. Statt existierende Ansätze zu verändern, bis 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 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 aus diesen Überlegungen resultierende Ansatz zur Überlastkontrolle wird dann auf den Fall eines Senders verallgemeinert, der dieselben Daten zeitgleich an mehrere Empfänger übertragen möchte, also auf die Multicast-Kommunikation. Im Anschluss daran wird aufgezeigt, wie durch die implizit koordinierte Kombination von Übertragungen mittels Network Coding die Übertragungskapazität erhöht werden kann. Abschließend wird eine Technik zur Offline-Zeitsynchronisation eingeführt, die eine zentrale Schwierigkeit bei der Auswertung experimenteller Messungen in drahtlosen Multihop-Netzwerken löst: Wegen abweichender Uhren lassen sich die von unterschiedlichen Netzwerkteilnehmern in ihren Ereignisprotokollen aufgezeichneten Zeitstempel nicht unmittelbar vergleichen; auch dieses Problem kann durch Ausnutzen des Wissens über die Struktur des Netzwerks gelöst werden.

2 Implizite Überlastkontrolle: CXCC

Die erste Fragestellung, die hier betrachtet werden soll, ist die Überlastkontrolle. Die Aufgabe eines Überlastkontrollmechanismus ist die Regelung der Datenrate, mit der ein Sender Daten in ein Netzwerk übertragen darf. Übersteigt die Rate der eingespeisten Daten die Kapazität des Netzwerkes signifikant, so kann dieses aufgrund der Überlast kollabieren.

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 höchst problematisch erwiesen [FML02, RK06]. 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 Paket-

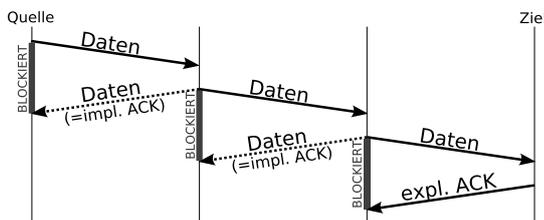


Abbildung 1: Paketweiterleitung in CXCC.

verluste auf, beispielsweise wegen sogenannter Kollisionen – zeitgleicher, sich gegenseitig störender Übertragungen – oder variierender Eigenschaften des Übertragungsmediums. Schon die Grundannahme der TCP-Überlastkontrolle, dass Paketverluste durch Pufferüberläufe vor einer überlasteten Verbindung 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 sehr 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 [SLM08] 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 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 zwischenzuspeichern. 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 erreichbaren 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. Diese Probleme lassen sich jedoch – wiederum unter Berücksichtigung der Besonderheiten der Umgebung – lösen. Details sind in [SLM08] beschrieben.

3 Implizite Zuverlässigkeit: BarRel

Sowohl umfangreiche simulative Untersuchungen als auch eine reale Implementation zeigen, dass CXCC – über ein großes Spektrum von Umgebungsparametern und bezüglich vieler verschiedener Metriken – im Vergleich zu konkurrierenden Ansätzen überragende Ergebnisse erzielt. Als großer Vorteil erweist sich dabei, dass für die Überlastkontrolle keine Kontrollpakete notwendig sind, die wie die Bestätigungspakete von TCP vom Ziel zurück zur Quelle laufen, denn solche Pakete wirken sich in drahtlosen Multihop-Netzwerken sehr nachteilig aus [dOB07]. Sie konkurrieren mit den Datenpaketen um das Medium und sind häufig die Ursache von Kollisionen und damit Paketverlusten. 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) [SKLM] konnte konstruktiv der Beweis erbracht 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 CXCC's 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 *Back-pressure Multicast Congestion Control* (BMCC) umgesetzt, dessen Details in [STL⁺07] 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 im Zusammenspiel mit dem geographischen Multicast-Routingprotokoll Scalable Position-Based Multicast (SPBM) [TFW⁺07] implementiert und untersucht.

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. Eine Vielzahl theoretischer Arbeiten zeigte bereits, dass dadurch große Kapazitätsgewinne möglich sind. Ein einfaches Beispiel in drahtlosen Netzwerken zeigt Abbildung 2. In Abbildung 2(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 2(b) werden die gleichen Pakete aus-

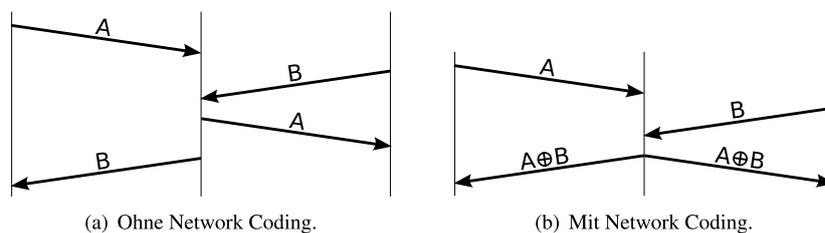


Abbildung 2: Paketaustausch mit und ohne Network Coding.

getauscht. 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 [KRH⁺06] 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 Ansatz *Near-Optimal Co-ordinated Coding* (noCoCo) [SHC07] ersetzt das rein opportunistische Vorgehen von COPE in Situationen, in denen gegenläufiger Datenverkehr dieselben Zwischenstationen passiert. Dabei werden die Möglichkeiten, die das drahtlose Multihop-Medium bietet, abermals konsequent genutzt, um implizit eine Koordination der Übertragungen benachbarter Netzwerkknoten zu erreichen. 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. Auch stellt sich heraus, dass es ideal mit der impliziten schrittweisen Überlastkontrolle in CXCC kombiniert werden kann. Vergleichende Untersuchungen mit COPE zeigen, dass sich der Datendurchsatz mit koordiniertem Network Coding oft um ein Vielfaches steigern lässt, und dass gleichzeitig die Paketlaufzeit und der durch Kontrolldaten und Übertragungswiederholungen verursachte Overhead stark vermindert werden.

6 Offline-Zeitsynchronisation

In den bislang vorgestellten Ansätzen CXCC, BarRel, BMCC und noCoCo wurden die Eigenschaften des Mediums genutzt, um im laufenden Betrieb Informationen implizit zu gewinnen und auf dieser Basis Protokolle zu entwickeln, die für drahtlose Multihop-

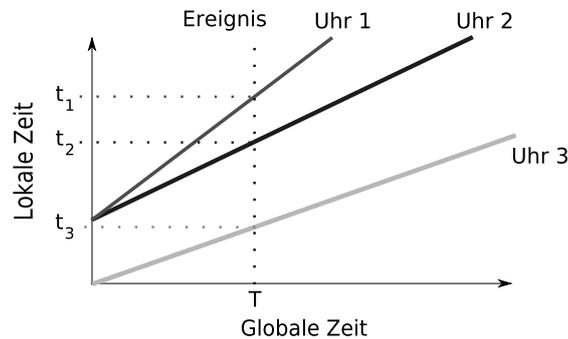


Abbildung 3: Zeitstempelvergabe mit mehreren lokalen Uhren.

Netzwerke besonders gut geeignet sind. Bei der experimentellen Evaluation solcher (und anderer) Protokolle stellt sich regelmäßig die Frage, wie auf Basis der von den Netzwerkknoten aufgezeichneten lokalen Ereignisprotokolle Schlüsse über das globale Geschehen im Netzwerk gezogen werden können. Das Problem hierbei ist, dass jeder Knoten zur Vergabe der Zeitstempel in seinem Ereignisprotokoll seine lokale Uhr verwendet. Diese Uhren weichen aufgrund von Gangungenauigkeiten sehr schnell in einer Größenordnung voneinander ab, die die benötigte Zeitaufösung bei weitem übersteigt. Der Einsatz von Zeitsynchronisationsprotokollen wie NTP [Mil94] kommt häufig nicht in Betracht, da hierdurch zusätzlicher Datenverkehr verursacht wird, der mit dem Datenverkehr des Experiments wechselwirken und so die Ergebnisse verfälschen kann.

Auf den ersten Blick unterscheidet sich diese Fragestellung stark von den bislang diskutierten Problemen. Dennoch stellt sich überraschenderweise heraus, dass ihre Lösung mit genau denselben Ansätzen möglich ist. Das drahtlose Multihop-Medium erlaubt es, ohne jeglichen zusätzlichen Datenverkehr implizit Informationen zu gewinnen, die für die Korrektur der Uhrenfehler genutzt werden können. Die Existenz und Effektivität einer solchen Lösung untermauert somit die Grundannahmen dieser Arbeit nochmals aus einer ganz anderen Perspektive – und löst darüber hinaus ein fundamentales Problem bei der experimentellen Untersuchung der zuvor diskutierten ebenso wie vieler anderer Ansätze.

Der hier vorgeschlagene Ansatz zur *Offline-Zeitsynchronisation* beruht darauf, die Ereignisprotokolle zunächst in den Knoten mit lokal abweichenden Uhren aufzuzeichnen, sie dann nach dem Experiment zusammenzuführen und erst im Nachhinein die Abweichungen der Uhren zu identifizieren und zu korrigieren. Hier kommt zum Tragen, dass Übertragungen auf dem drahtlosen Medium von mehreren Knoten empfangen werden können. Da diese Empfangsvorgänge nahezu zeitgleich erfolgen, können sie genutzt werden, um die lokalen Uhren der Empfänger zueinander in Beziehung zu setzen. So liefert der im Experiment untersuchte Datenverkehr selbst die zur Korrektur notwendige Information. Wird beispielsweise wie in Abbildung 3 eine Übertragung von drei Knoten empfangen, und werden lokal hierfür die Zeiten t_1 , t_2 und t_3 protokolliert, so lässt sich auf dieser Basis beim Zusammenführen der Ereignisprotokolle nachvollziehen, dass t_1 , t_2 und t_3 denselben realen Zeitpunkt T repräsentieren.

Das Ausnutzen dieses Wissens wird jedoch durch zwei Tatsachen erschwert. Zunächst erfolgt die Aufzeichnung in den Knoten oft nicht unmittelbar, sondern mit einer nichtdeterministischen Aufzeichnungsverzögerung, sodass die lokalen Zeitstempel nicht als strikt zeitgleich angenommen werden dürfen, sondern nur Anhaltspunkte liefern. Außerdem werden Übertragungen in der Regel nicht von *allen* Knoten im Netzwerk, sondern nur von ständig wechselnden Teilgruppen empfangen. Aufgrund dieser Umstände stellt das Korrigieren der lokalen Uhrenfehler ein mathematisch anspruchsvolles Problem dar.

Auf Basis einer Reihe von Annahmen über das Verhalten der Uhren und die Aufzeichnungsverzögerung lässt sich ein Maximum-Likelihood-Schätzer für die Korrektur formulieren. Dieser kann in ein lineares Optimierungsproblem transformiert werden. Bedingt durch seine Größe von typischerweise mehreren hunderttausend Unbekannten und Millionen von Nebenbedingungen ist dessen Lösung allerdings nicht trivial. Wie sich herausstellt, kann aber die spezielle Struktur des Problems genutzt werden, um es mittels eines speziell angepassten Lösungsverfahrens auf Basis von Mehrotras Predictor-Corrector-Algorithmus [Meh92] – einem Innere-Punkte-Verfahren – höchst effizient zu lösen. Analytische und simulative Evaluationen sowie Messungen mit realer Hardware demonstrieren die äußerst hohe Präzision, die mit dieser Vorgehensweise erreicht werden kann. Für Details zum Ansatz ebenso wie zu den Resultaten sei hier auf [SKR⁺] verwiesen.

7 Zusammenfassung

Den Kern dieser Arbeit bilden eine Reihe von Lösungsansätzen für verschiedenartige Probleme in drahtlosen Multihop-Netzwerken. Alle haben 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 komponieren oder zu kaschieren, werden die Besonderheiten der spezifischen Umgebung als etwas Positives aufgefasst, das selbst zu effektiven Lösungen beitragen kann.

Jeder der vorgeschlagenen Ansätze nutzt Wissen über die Eigenschaften des Netzwerks, um Informationen implizit zu gewinnen, deren Beschaffung sonst sehr aufwändig oder gar unmöglich wäre. Sie alle – die Protokolle CXCC, BarRel, BMCC und noCoCo ebenso wie die Offline-Zeitsynchronisation – zeigen, dass dadurch neue und manchmal unkonventionelle, doch höchst effektive Lösungen möglich werden.

Literatur

- [dOB07] Ruy de Oliveira und Torsten Braun. A Smart TCP Acknowledgment Approach for Multihop Wireless Networks. *IEEE Transactions on Mobile Computing*, 6(2):192–205, Februar 2007.
- [FML02] Zhenghua Fu, 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.

- [KRH⁺06] Sachin Katti, 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.
- [Meh92] Sanjay Mehrotra. On the Implementation of a Primal-Dual Interior Point Method. *SIAM Journal on Optimization*, 2(4):575–601, 1992.
- [Mil94] David L. Mills. Internet Time Synchronization: The Network Time Protocol. In Zhonghua Yang und T. Anthony Marsland, Hrsg., *Global States and Time in Distributed Systems*. IEEE Computer Society Press, 1994.
- [RK06] Vivek Raghunathan und Panganamala R. Kumar. A Counterexample in Congestion Control of Wireless Networks. *Elsevier Performance Evaluation*, 64:399–418, Juni 2006.
- [SHC07] Björn Scheuermann, Wenjun Hu und Jon Crowcroft. Near-Optimal Co-ordinated Coding in Wireless Multihop Networks. In *CoNEXT '07: Proceedings of the 3rd International Conference on Emerging Networking Experiments and Technologies*, Dezember 2007.
- [SKLM] Björn Scheuermann, Markus Koegel, Christian Lochert und Martin Mauve. Reliable Wireless Multihop Communication without End-to-End Acknowledgments. Zur Veröffentlichung eingereicht.
- [SKR⁺] Björn Scheuermann, Wolfgang Kiess, Magnus Roos, Florian Jarre und Martin Mauve. On the Time Synchronization of Distributed Logfiles in Networks with Local Broadcast Media. *IEEE/ACM Transactions on Networking*. Im Druck.
- [SLM08] Björn Scheuermann, 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.
- [STL⁺07] Björn Scheuermann, 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.
- [TFW⁺07] Matthias Transier, 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.



Björn Scheuermann studierte ab 2000 Mathematik und Informatik an der Universität Mannheim. Er schloss sein Studium 2004 mit einem Bachelor of Science in Mathematik und Informatik sowie einem Informatik-Diplom ab. Er ging 2005 als wissenschaftlicher Mitarbeiter an den Lehrstuhl für Rechnernetze und Kommunikationssysteme der Heinrich-Heine-Universität Düsseldorf. Im Frühjahr 2007 besuchte er für einen Forschungsaufenthalt das Computer Laboratory der University of Cambridge. Er promovierte 2007 mit dem Thema „Reading Between the Packets – Implicit Feedback in Wireless Multihop Networks“ und erhielt die Gesamtnote „summa cum laude“. Seither ist er

als promovierter Wissenschaftler weiter in Düsseldorf tätig. Seine Forschungsinteressen liegen in den Bereichen Mobile Netzwerke und Fahrzeug-zu-Fahrzeug-Kommunikation.