

Semesterarbeit

Messungen in Ad-Hoc-Netzwerken

André Bayer

bayeran@student.ethz.ch

Departement Informatik
Eidgenössische Technische Hochschule Zürich
30. August 2003

Prof. Dr. Roger Wattenhofer
Distributed Computing Group
Betreuer: Aaron Zollinger

Inhaltsverzeichnis

1 Aufgabenstellung	3
2 Motivation	3
3 Messungen	4
4 Rejoin	4
4.1 Windows XP	4
4.2 Linux	5
5 Vollständiger Graph	5
5.1 2 Peers	5
5.2 Mehrere Peers	6
5.3 Round Trip Time (RTT)	7
5.4 Verschiedene Distanzen	7
5.5 Position	8
6 Multihop	9
7 Wo gehen Pakete verloren	9
7.1 Messungen	9
7.2 Backoff-Algorithmus	10
8 Fazit	11
9 Persönliche Erfahrungen	11
10 Referenzen	12

1 Aufgabenstellung

In dieser Semesterarbeit geht es um Tests und Messungen mit Wireless LAN(IEEE 802.11b)-Karten im Ad-Hoc Modus. Die Aufgabenstellung umfasste zu Beginn folgende Fragen:

- Funktionieren die getesteten Karten grundsätzlich erwartungsgemäss (zum Beispiel nach Entfernen und Zusammenführen von Geräten)?
- Gibt es Begrenzungen in Bezug auf
 - Anzahl Geräte in komplett vermaschten Netzen?
 - Länge einer Multihop-Kette?
- Gibt es Probleme bei der Kommunikation zwischen Karten des einen Herstellers, bei solchen eines anderen jedoch nicht?
- Treten Kompatibilitätsprobleme zwischen Karten unterschiedlicher Hersteller auf?
- Gibt es Unterschiede/Probleme mit Windows und Linux?

Ich führte alle Messungen mit zwei verschiedenen Modellen von WLAN-Karten durch. Das waren die in den Neptun Laptops eingebaute IBM-Karte und die Cisco-Karte, die über den Studentendiscount [1] zu günstigen Konditionen vertrieben wird und unter den Studenten weit verbreitet ist. Zu Beginn war noch offen, ob während dem Semester noch weitere Karten angeschafft werden. Das wichtigste Kriterium an die neuen Karten war, dass die Sendeleistung manuell eingestellt werden konnte, damit die Reichweite variabel ist. Leider konnte mir kein Hersteller Support darauf eine positive Antwort geben, weshalb ich es dann bei den zwei Karten belies.

Während der Arbeit veränderte sich die Aufgabenstellung leicht. Ein Schwerpunkt war die Frage, wie gut funktioniert die Paketübertragung in Ad-Hoc Netzen, also wie hoch ist der Paketverlust je nach Anzahl von Laptops. Ausserdem war die Frage: Wie lange dauert die Übertragung, gemessen mit der Round Trip Time (RTT). Am Schluss beschäftigte ich mich noch mit der Frage, wo und warum Pakete verloren gehen.

2 Motivation

Ich besuchte im Sommersemester 02 die Vorlesung „Mobile Computing“ bei Professor Wattenhofer [2]. In den Übungsstunden entwickelten wir einen Instant Messenger, um in Ad-Hoc Netzen zu chatten. Leider war der Kommunikationsgraph sehr instabil, d.h. Knoten verschwanden plötzlich und tauchten wieder auf. So war es fast unmöglich geordnet zu kommunizieren und Pakete über dieses Netz zu routen. Aus diesem Problem heraus entstand diese Semesterarbeit. Ad-Hoc Netze interessierten mich sehr und als ich diese Arbeit ausgeschrieben sah, war für mich klar, dass das eine spannende Semesterarbeit für mich sein würde.

3 Messungen

Um die Messungen durchzuführen, habe ich ein Programm entwickelt, das auf dem Prinzip des Instant Messengers der Mobile Computing-Vorlesung aufbaut. Für die Kommunikation wird UDP verwendet und die Pakete werden über eine Multicast Adresse versendet. Es gibt zwei Arten von Paketen, Ping und Pong. Ping-Pakete werden per Broadcast versendet und alle Geräte, die so ein Paket erhalten, schicken ein Pong-Paket an den Versender zurück. So kann gezählt werden, von welchem Gerät wie viele der gesendeten Pakete zurückkommen. Daraus kann die Verlustrate der Pakete berechnet werden, was ich für die meisten Versuche brauchte. Im Laufe der Arbeit habe ich das Programm weiterentwickelt und eine Funktion zum Messen der Round-Trip-Time hinzugefügt. Eine andere Erweiterung ist das Zählen der Ping-Pakete und daraus die Berechnung der Verlustrate der Pings. Somit konnte ich sehen, ob eher die Pings oder die Pongs verloren gehen.

Eine zusätzliche Version des Programms unterstützt Flooding mit Dublikaterkennung, womit ich eine einfache Routingfunktion hatte, um die Multihop-Kette zu analysieren.

4 Rejoin

Bei den ersten Tests, die ich durchführte ging es darum, die Funktionalität der Karten zu überprüfen. Vor allem ging es mir um die Frage, was passiert, wenn zwei Geräte entfernt und wieder zusammengeführt werden. Ich nenne diesen Test „Rejoin“.

Wenn man sich mit einem Laptop von einem anderen Sender immer weiter entfernt, dann gibt es eine Schwelle, bis dahin praktisch alle Pakete empfangen werden und danach keine mehr.

Die Distanz des Empfangsbereichs hängt sehr von den baulichen Gegebenheiten ab und ist grösser bei Sichtkontakt, als wenn dazwischen Mauern sind. Bei meinen Messungen im HRS-Gebäude war die Empfangsweite der Cisco Karte 20m, resp. 30m bei der IBM-Karte.

Erstaunlich ist, dass die Signalstärke nicht bei 0% ist, wenn keine Pakete mehr empfangen werden. Stattdessen liegt die Schwelle bei der IBM Karte bei 20% Signalstärke und bei der Cisco Karte ist sie sogar schon bei 40%, auch wenn die Signalqualität noch etwa 70% ist.

Die Signalstärke gibt an, wie gut das Sendesignal empfangen wird. Die Signalqualität bedeutet, wie stark das Signal durch Rauschen oder Interferenzen verzerrt wird. In der Cisco- oder Windowsstatusanzeige werden beide als relative Wert angezeigt.

Die Tests habe ich auf zwei Arten durchgeführt:

- Der mobile Laptop wird vom Sender entfernt, bis die Schwelle erreicht ist, es wird 10 Sekunden gewartet und dann der Laptop wieder in den Empfangsbereich zurück gebracht.
- Der mobile Laptop wird vom Sender entfernt, bis das Signal ganz weg ist, es wird 1 Minute gewartet und dann der Laptop wieder zurück gebracht.

Dabei habe ich Unterschiede zwischen Windows und Linux festgestellt.

4.1 Windows XP

Der kurze Unterbruch des Empfangssignals stellt kein Problem dar, denn nach wenigen Sekunden sind die zwei Karten wieder synchronisiert und können Pakete empfangen. Der lange Unterbruch mit dem vollständig verlorenen Signal führt dazu, dass die Geräte sich ganz verlieren. Das System bringt die Meldung „Network unavailable“, wodurch auch die IP-

Adresse gelöscht wird. Wieder im Empfangsbereich wird die IP neu gesetzt. Die IBM-Karten finden sich gar nicht mehr und sie müssen ganz neu gestartet werden. Bei den Cisco-Karten dauert es ungefähr eine Minute, bis die Kommunikation wieder normal läuft. Das ist hauptsächlich die Zeit, welche vergeht, bis die IP-Adresse dynamisch erteilt ist. Falls eine feste IP verwendet wird, finden sie sich schneller wieder.

Wird eine IBM und eine Cisco Karte gemischt, dann klappt der Rejoin auch. Die zwei Karten finden sich nach einer kurzen Zeit wieder, gleich wie wenn zwei Cisco Karten verwendet werden.

4.2 Linux

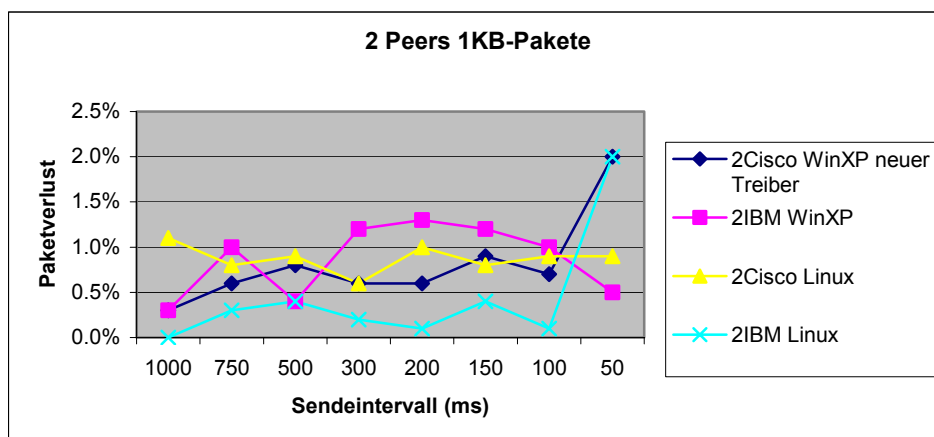
Unter Linux funktionieren beide Tests ohne Probleme. Nach einer kurzen Synchronisationszeit können die Karten wieder kommunizieren. Speziell bei dem Test war, dass die Karten ohne Power-Save-Mode liefen, was dazu führte, dass sie immer weiter sendeten, auch wenn keine andere im Empfangsbereich war. Die Netzwerkverbindung ging deswegen auch nicht verloren, was dazu führt, dass sie sich schnell wieder finden.

Mit der IBM Karte unter Windows kann der Power-Save-Modus abgeschaltet werden, was für die zwei Tests aber nichts brachte. Es gab zwar keinen Netzunterbruch mehr und die Pings wurden weiter versendet, aber die Karten können sich trotzdem nicht synchronisieren und müssen neu gestartet werden. Für die Cisco Karte gibt es drei verschiedene Power-Save-Modi. Bei diesen Tests habe ich aber damit keinen Unterschied festgestellt.

5 Vollständiger Graph

Bei diesen Messungen hören sich alle Peers gegenseitig, sie sind darum zu einem kompletten Graph vermascht. Alle Geräte senden Pings und antworten, auf alle empfangenen Ping-Pakete, mit einem Pong. Die Tests beginnen mit zwei Laptops und dann wird laufend ein Weiterer gestartet, um zu sehen, ob es eine obere Grenze für die Anzahl gibt. Leider hatte ich maximal 7 Laptops zur Verfügung und bis da gab es keine Probleme. Ich denke aber auch nicht, dass mit 10 oder 15 Laptops eine Grenze erreicht wird, sondern dass diese höher liegt. In [3] wird beschrieben, dass ab 30 Stationen Probleme mit der Zeitsynchronisation auftreten können. Anders sieht es aus, wenn sehr viele Pakete versendet werden.

5.1 2 Peers



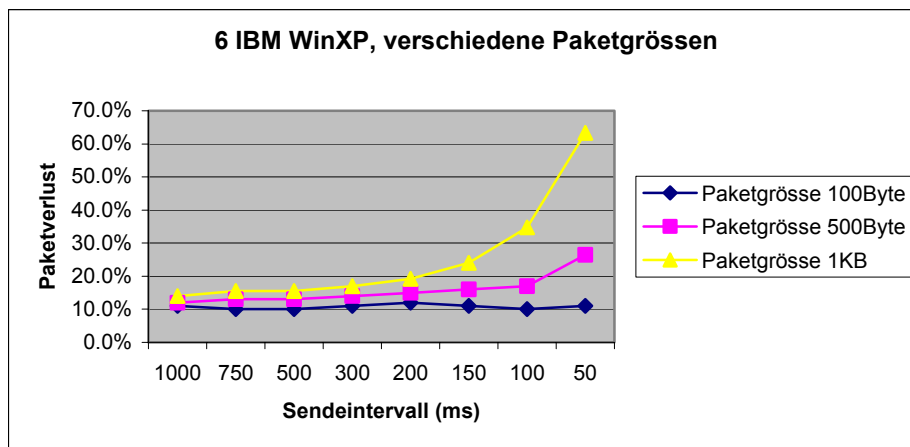
Die Messungen habe ich mit 4 verschiedenen Konfigurationen durchgeführt. Mit der Cisco und der IBM Karte jeweils unter Windows und Linux. Beide Peers senden mit dem gegebenen

Intervall (alle 1000ms bis 50ms, x-Achse) Ping Pakete. Gezählt wird dann, wie viele der retournierten Pongs ankommen. Relativ zu der Anzahl Pings, die versendet wurden, gibt das eine Paketverlustrate, die als y-Achse im Diagramm dargestellt ist.

Die Cisco Karte unter Windows fiel zuerst stark ab gegenüber den anderen Konfigurationen. Aus diesem Grund installierte ich einen neuen Treiber [4], was dazu führte, dass alle Konfigurationen ungefähr gleich gut sind. Dies ist auch im Diagramm gut zu erkennen. Die Paketverlustrate um 1% zeigt, dass der Ad-Hoc Modus für 2 Peers gut funktioniert.

5.2 Mehrere Peers

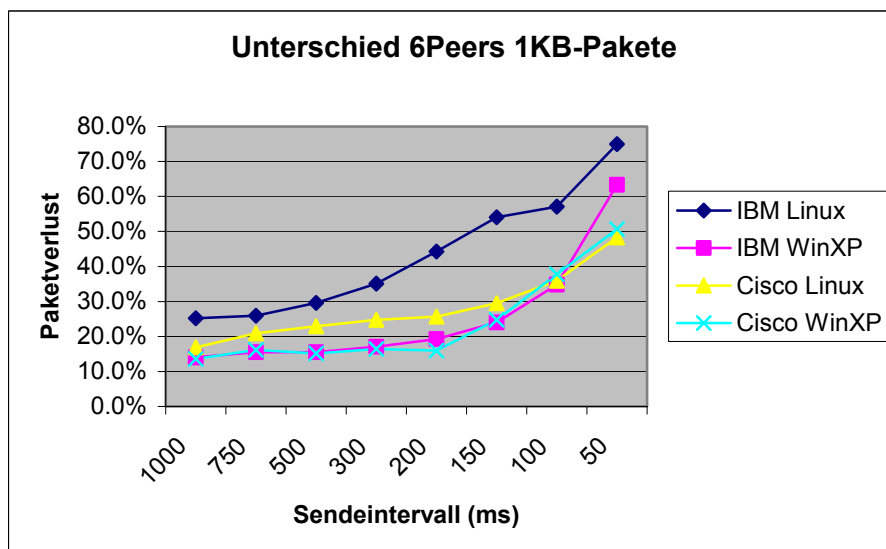
Mit der Erhöhung der Anzahl Laptops steigt auch die Verlustrate der Pakete. Ein zweiter wichtiger Faktor ist die Grösse der Pakete. Für meine Messungen habe ich 100 Byte oder 1 Kilobyte grosse Pakete verwendet. Vor allem 1KB schien eine gute Grösse zu sein und die



Messungen ergaben damit die schönsten Resultate. Im Diagramm sieht man, dass bei 6 Peers auch mit einer kleinen Netzbelastung die Verlustrate schon bei 10% liegt. Mit kleinen Paketen bleibt der

Paketverlust auch bei sehr vielen Paketen gleich gross. Allerdings steigt der Paketverlust mit 1KB Paketen mit kleinen Intervallen stark an und wenn alle 50ms ein Paket versendet wird, liegt er über 60%. Dies entspricht einer theoretischen Netzbelastung von 50% bei einem 11MBit Netz. Die von Windows gemessene Belastung liegt aber nur bei 18%, was einer Paketverlustrate von 64% entspricht. Dies entspricht ziemlich genau dem Wert, den ich mit meinem Programm gemessen habe.

Die vier verschiedenen Konfigurationen im Vergleich zeigen, dass bei allen die Verlustrate mit 6 Peers ziemlich hoch ist. In diesem Vergleich fällt die IBM-Karte unter Linux gegenüber den anderen etwas ab. Allerdings kann man nicht daraus folgern, dass die IBM-Karte

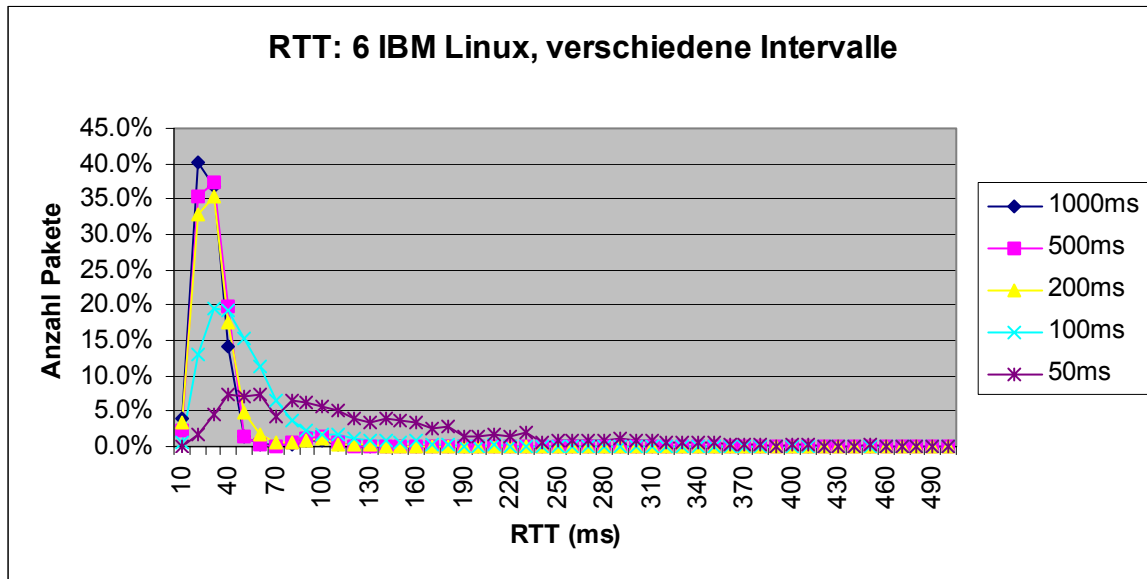


schlechter ist. Vielmehr kommt es darauf an, welcher Treiber verwendet wird. Die Werte der Cisco Karte verbesserten sich unter Windows mit dem neuen Treiber um ca. 20%. Daran sieht man auch, dass noch Verbesserungsmöglichkeiten vorhanden sind. Ich hatte allerdings nur

einen Treiber für die IBM-Karte, bei dem unter Linux in den Ad-Hoc Modus gewechselt werden kann [5]. Darum konnte ich keine Vergleichsmessung durchführen, um zu sehen, ob mit einem anderen Treiber ein besseres Resultat erzielt worden wäre.

5.3 Round Trip Time (RTT)

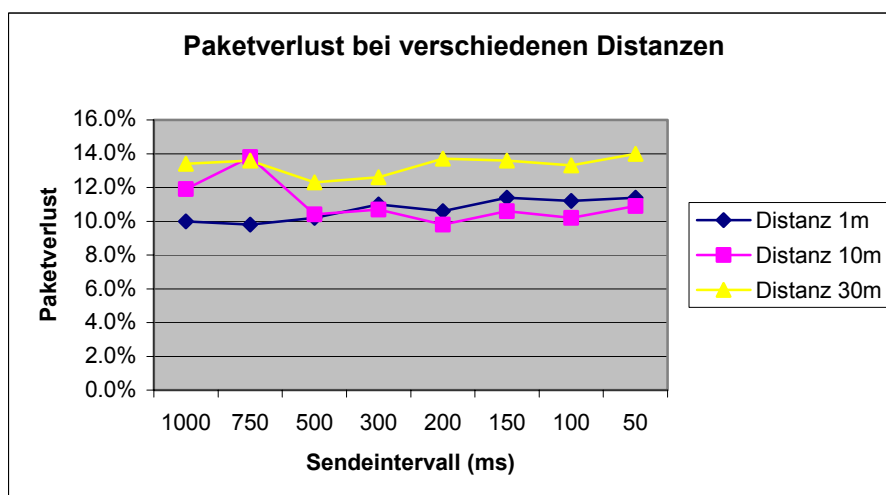
Als RTT habe ich die Zeit gemessen, die zwischen dem Verschicken eines Pings bis zum Empfang des dazugehörigen Pongs, vergeht. Aus den Daten habe ich den Durchschnitt und die Standardabweichung berechnet und als Grafik aufgezeigt, wie viele Messungen prozentual in welchem 10ms-Intervall liegen. Bei kleiner Netzbelastung sind typische Werte für den



Durchschnitt 20ms und für die Standardabweichung 40ms. Vereinzelt gibt es aber auch Pakete, die bis zu einer Sekunde unterwegs sind. Vor allem, wenn die Netzbelastung steigt, verschmiert sich die Kurve über das ganze Spektrum.

5.4 Verschiedene Distanzen

Bis jetzt haben bei allen Messungen die Laptops einen Abstand von höchstens 4m gehabt. Das hat folgende Gründe. Die Distanz hat einen kleinen Einfluss auf die Ergebnisse und ich konnte die Messungen im Labor durchführen. Meine Laptops standen so nur selten unbeaufsichtigt im HRS Gebäude und es ist auch nicht ganz einfach, 5 Laptops über eine

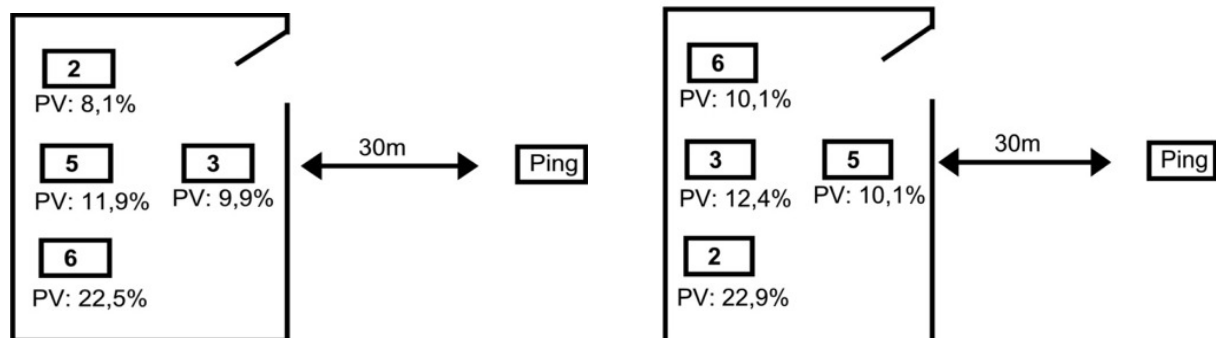


Etage verteilt zu bedienen. Um die Auswirkung von verschiedenen Distanzen zu untersuchen, habe ich einen Laptop verschieden weit von vier anderen aufgestellt und von diesem aus Pings verschickt. Die anderen vier schicken nur Pongs

zurück. Die Kurven der RTT Messungen sehen bei allen Distanzen identisch der oberen Grafik aus. Die Entfernung von einzelnen Peers, die bei wireless Ad-Hoc Netzen erfahrungsgemäss nicht so gross ist, hat also auf die RTT keinen Einfluss. Weil die Ausbreitungsverzögerung klein ist, verglichen mit der Übertragungsverzögerung. Die Verarbeitung im Gerät selber ist also entscheidend, wie lange die Übertragung dauert. Die Paketverlustrate steigt mit der Distanz leicht an. Wie schon weiter oben (Kapitel Rejoin) erwähnt, gibt es in einer gewissen Distanz eine Schwelle, bis wohin die Pakete empfangen werden und darüber nicht mehr. In kurzer Distanz liegt die Fehlerrate ungefähr bei 10% und in der Maximalen, also kurz vor der Schwelle, bei 14%.

5.5 Position

Als ich die Messungen mit verschiedenen Distanzen durchführte, fiel auf, dass mit grösserer Distanz die Unterschiede in der Fehlerrate der einzelnen Peers stärker wurden. Der Paketverlust des einen Peers war sehr klein und der eines anderen sehr gross. Diese Beobachtung zog sich durch alle Messungen hindurch und es waren immer die gleichen, die besser waren. Das liess darauf schliessen, dass die Position der einzelnen Laptops eine entscheidende Rolle spielt, ob seine Ping Pakete ankommen oder nicht. Vor allem bei der IBM-Karte unter Windows war der Effekt gut zu beobachten.



Bei der oberen Versuchsanordnung stehen die vier Laptops 2, 3, 5, 6 im Laborraum und ein Laptop ausserhalb in 30m Entfernung. Als räumliche Gegebenheit ist zu beachten, dass zwischen Laptop 2 und 3 eine Tür ist (im linken Bild). Deswegen ist die Position 6 am Weitesten vom Ping-Rechner entfernt. Vom linken zum rechten Bild wurden die Laptops 2 und 6, 3 und 5 in ihren Positionen vertauscht und derselbe Versuch nochmals durchgeführt. Die Position 6 im linken Bild ist eindeutig benachteiligt und wird von den anderen drei wie abgeschirmt, so dass seine Ping-Pakete öfters verloren gehen. Ich habe aber nicht genügend Versuche durchgeführt, um sagen zu können, welche Anordnungen oder Positionen Probleme verursachen. Es ist auch sehr schwierig, die Messungen zu reproduzieren, denn wenn ein Laptop um ein paar Zentimeter verschoben wird, können die Werte ganz anders aussehen. Das Ergebnis dieses Versuches ist, dass die Position der Peers einen Einfluss auf die Paketverlustrate hat.

6 Multihop

Ich hatte in meinem Testprogramm nur ein einfaches Flooding mit Dublikaterkennung implementiert und darum mussten die Laptops genau platziert werden, um eine Multihop-Kette zu haben. Ein mittlerer Peer sollte seine beiden Nachbarn empfangen, diese dürfen sich jedoch gegenseitig nicht hören, damit beim Routing kein Peer übersprungen wird. Das erwies sich schwieriger zu erreichen, als ich zuerst gedacht hatte. Aus Zeitgründen belies ich es deshalb bei einigen wenigen Messungen.

Als Ergebnis kam heraus, dass ich in Bezug auf die Anzahl keine Limite festgestellt habe. Wie erwartet, stieg die Paketverlustrate mit jedem zusätzlichen Hop an. Je nach Versuch lag die Zunahme zwischen 5- 10%, also relativ hoch. Wenn mit zusätzlichen Peers redundante Wege in den Graph eingebaut werden, kann der Paketverlust ziemlich tief gehalten werden. Die Netzbelastung steigt zwar, dafür können einzelne Paketverluste ausgeglichen werden. In kompletten Graphen bringt das Flooding auch etwas, die Verlustrate liegt zwischen 5 und 25%, jedoch steigt die Netzbelastung enorm schnell an. Vergleichswerte mit 6 Cisco-Karten und alle senden Pings: Verlustrate zwischen 13 und 50% (siehe 5.2 Mehrere Peers).

Bei der Cisco Karte kann die Sendeleistung eingestellt werden. Aber gerade mit der kleinsten Sendeleistung, die ideal wäre um Tests mit Multihop Ketten durchzuführen, ist die Paketverlustrate am grössten.

Einen Unterschied zwischen der IBM und der Cisco Karte festzustellen ist schwierig, weil sie verschiedene Reichweiten haben, können die Laptops nicht an denselben Stellen platziert werden. So ist ein Vergleich schwierig.

7 Wo gehen Pakete verloren

Die Beobachtung, dass die Position eine wichtige Rolle spielt, lässt die Frage aufkommen, ob die Pings gar nicht ankommen oder ob erst die Pongs verloren gehen. Da beim Versenden eines Pings praktisch alle dieses Paket gleichzeitig erhalten und darauf mit einem Pong reagieren, könnte das gleichzeitige Versenden der Pakete von mehreren Peers aus ein Problem sein, wenn sich die Pakete gegenseitig auslöschen. Auf den Punkt gebracht interessiert die Frage, ob mehr Pings oder Pongs verloren gehen und ob Massnahmen gegen den Paketverlust getroffen werden können.

7.1 Messungen

Die Messungen habe ich mit der einfachsten Übertragung begonnen, nämlich mit 2 Laptops, wovon einer Pings schickt. Als zweiter Versuch schickt einer einen Ping und der andere antwortet mit einem Pong. Im nächsten Versuch sendet wieder nur einer, aber 4/5 antworten mit einem Pong. Als letztes schicken alle 6 Laptops Ping- und Pong-Pakete, wie bereits bei den Messungen zum vollständigen Graph.

Die Daten sind Durchschnittswerte von allen durchgeführten Messungen zu den jeweiligen Tests. Alle Messungen wurden unter Windows XP durchgeführt. Bei den Cisco Karten habe ich zwischen neuem und altem Treiber unterschieden, weil die Ergebnisse recht unterschiedlich waren. Die vier untersten Messungen führte ich nicht, wie die anderen mit allen Intervallen durch, von 1000ms bis 50ms, sondern nur für einen spezifischen Wert. Als erstes mit 100 Byte Paketen im 1000ms Intervall und danach mit 100 Byte und 100ms.

Über alle Messungen gesehen kann gesagt werden, dass nicht wesentlich mehr Pong-Pakete verloren gehen als Pings. Eine Ausnahme ist bei der IBM-Karte bei einem Ping- und vier Pong-Clients. Dort ist der zweite Wert um einiges höher als der erste.

Die Ergebnisse zu allen Tests sind in der unten stehenden Tabelle aufgelistet.

Test	Bemerkung	Pingverlust	Pongverlust
2 Cisco, nur Ping	neuer Treiber	0,1%	
2 IBM, nur Ping		0,6%	
2 Cisco, Ping/Pong	neuer Treiber	0,1%	0,4%
2 IBM, Ping/Pong		0,8%	1,1%
6 Cisco, 1 Ping Client	neuer Treiber	1,2%	
6 Cisco, 1 Ping & 5 Pong Clients	neuer Treiber	0,9%	2,7%
6 Cisco, 1 Ping & 5 Pong Clients	alter Treiber	8,0%	12,0%
6 Cisco, 1 Ping & 5 Pong Clients, Backoff	neuer Treiber	0,9%	2,4%
6 Cisco, 1 Ping & 5 Pong Clients, Backoff	alter Treiber	6,5%	9,3%
5 IBM, 1 Ping & 4 Pong Clients		1,1%	8,0%
5 IBM, 1 Ping & 4 Pong Clients, Backoff		1,2%	1,6%
6 Cisco, alle Ping/Pong	100B, 1000ms	4,5%	7,0%
6 Cisco, alle Ping/Pong, Backoff	100B, 1000ms	5,4%	6,5%
6 Cisco, alle Ping/Pong	100B, 100ms	7,8%	10,7%
6 Cisco, alle Ping/Pong, Backoff	100B, 100ms	30,0%	51,82%

7.2 Backoff-Algorithmus

Eine mögliche Ursache für einen Paketverlust ist das fast gleichzeitige Versenden der Pongs, wenn mehrere Peers aktiv sind. Um dieses „gleichzeitige Versenden“ zu umgehen, habe ich einen Backoff-Algorithmus in mein Programm eingebaut. Dieser Algorithmus nimmt eine zufällige Zeit zwischen 0 und 50ms, die vom Empfang eines Pings, bis das Pong zurückgesendet wird, gewartet wird. Alle Messungen, die ich mit dieser Version des Programms durchgeführt habe, sind in der obigen Tabelle, unter Punkt 7.1 mit „Backoff“ gekennzeichnet.

Bei der Cisco-Karte verbesserte sich das Ergebnis leicht. Allerdings mit der Wirkung des neuen Treibers verglichen, kann dies vernachlässigt werden. Daraus kann angenommen werden, dass im neuen Treiber das Warten auf einen freien Kanal einiges besser funktioniert als im alten.

Der grosse Pongverlust bei der IBM-Karte konnte stark gesenkt werden. Zusammen mit der Beobachtung, dass bei dieser Karte die Position einen grossen Einfluss auf den Paketverlust hat, kann man schliessen, dass sich die Pongs gegenseitig stark stören, also das Horchen auf einen freien Kanal nicht gut funktioniert. Dieses Ergebnis muss aber relativiert werden, denn ich habe die Messungen mit dem standardmässig installierten Treiber durchgeführt. Ob mit einem neuen Treiber dieses Problem schon behoben ist, kann ich nicht sagen, dazu müssten alle Versuche nochmals nach einem Treiberupdate durchgeführt werden.

Der grosse Nachteil des Backoff Algorithmus wird in der letzten Messung ersichtlich. Bei einer etwas grösseren Netzbelastung steigt der Paketverlust deutlich an. Das Problem liegt darin, dass zu lange gewartet wird, bis die Pong Pakete zurück gesendet werden und schon bald der Puffer überläuft und Pakete weggeworfen werden. Abhilfe könnte eine Verbesserung des Algorithmus bringen, indem die zufällige Wartezeit mit steigender Netzlast kleiner wird.

8 Fazit

Der Ad-Hoc-Modus der zwei getesteten WLAN Karten funktioniert mit zwei Laptops sehr gut. Jedoch steigt die Paketverlustrate mit der Anzahl aktiver Peers an. Ein zweiter Faktor ist die Bitrate eines Senders. Bei grösseren Paketen und kleineren Intervallen steigt der Paketverlust stark an. Eine höhere Belastung als 20% wird selten erreicht, weil der Verlust so gross wird, dass die darüber hinaus versendeten Pakete keine zusätzliche Belastung sind. Die Distanz spielt keine grosse Rolle, viel entscheidender ist die Position, an dem sich das Gerät befindet. Mit dem Backoff-Algorithmus konnten bei kleiner Netzbelastung bessere Ergebnisse erzielt werden. Bei hoher Belastung, wirkt sich der Algorithmus negativ aus. Die besten Verbesserungen werden aber mit einem Treiberupdate erzielt und das ist meiner Meinung nach auch ein Ort, wo noch Verbesserungsmöglichkeiten vorhanden sind. Die Karten verursachen öfters und meist unerklärlich Probleme. Beispielsweise gibt es oft kurze Aussetzer, die ab und zu bis zu einem Netzunterbruch führen, bei dem einige Sekunden nichts übertragen werden kann. Ich hatte auch Fälle, in denen die Karte ganz abstürzte und ein Neustart der Karte nötig war.

9 Persönliche Erfahrungen

Das war eine eher praktische Arbeit und ich hatte am Anfang das Gefühl, dass diese Messungen keine grösseren Probleme verursachen können. Ich musste dann aber bald einsehen, dass die Konfiguration der Laptops und das Einrichten einer neuen Messung sehr viel Zeit in Anspruch nimmt. Vor allem unter Linux bekundete ich Schwierigkeiten, die Karten in den Ad-Hoc-Modus zu bringen, denn ich arbeite sonst meist mit Windows. Während der Arbeit lernte ich sehr viel über Netzwerkeinstellungen, Treiberinstallation und Konfigurationen unter Linux, die für mich eher ungewohnt allesamt im Terminal vorgenommen werden. Mein Assistent und andere Studenten im Labor zeigten sich sehr hilfsbereit und gaben mir nützliche Tipps. Ich möchte hiermit allen danken, die mich während der Arbeit unterstützt haben.

Es war auch nicht immer ganz einfach, die Messungen durchzuführen, weil oft irgendetwas nicht klappte und keine Pakete übertragen wurden. Die Frage war dann: Geht das jetzt nicht oder habe ich eine Einstellung falsch gewählt. Mit genügend Geduld und Ausprobieren funktionierte es dann doch und manchmal klappte es auch plötzlich, ohne dass ich eingreifen musste.

Das Unangenehmste, das passierte, war als ich den Treiber der Cisco Karte updatete und gleichzeitig eine neue Firmware auf die Karte geladen wurde. Unter Windows funktionierte sie zwar nachher viel besser, nur leider vertrug sich der Linuxtreiber mit der neuen Firmware nicht und sie konnte in diesem System nicht mehr gestartet werden.

Alles in allem war es eine interessante Arbeit und ich konnte mich intensiv mit dem Ad-Hoc Modus beschäftigen. Auch wenn nicht alle Ergebnisse so raus kamen wie ich es erhoffte, kann ich sagen, dass ich mich jetzt damit besser auskenne. Für die Zukunft wird es auch spannend sein, dieses Gebiet weiter zu verfolgen, denn in den PC Heften [6] wird vermehrt darüber geschrieben, wie zum Beispiel Ad-Hoc-Netze mit Access-Point in der Praxis als Mesh Networks realisiert werden.

10 Referenzen

- [1] Stiftung Studenten Discount
<http://www.ssd.ethz.ch>
- [2] Vorlesung Mobile Computing, Prof. R. Wattenhofer
<http://dcg.ethz.ch/lectures/mobicomp/index.html>
- [3] L. Huang, T-H. Lai. On the Scalability of IEEE 802.11 Ad Hoc Networks,
Juni 2002
- [4] Neuer Cisco Aironet Wireless Treiber, Cisco Software Center
<http://www.cisco.com/public/sw-center/sw-wireless.shtml>
- [5] Ad-Hoc fähiger Linux Treiber für die IBM Karte, Jason's Web Thingy
http://www.trekweb.com/~jasonb/articles/hostap_20021012.shtml
- [6] Mesh Networks, PC Professionell 9/03, August 2003, Seiten 126 - 130
<http://www.vnunet.de/pcpro/default.asp>